Università degli Studi di Bologna

Facoltà di Ingegneria

Corso di Laurea in Ingegneria Informatica

DESCRITTORI E METRICHE PER IL TRACCIAMENTO DI OGGETTI IN SEQUENZE DI IMMAGINI PER APPLICAZIONI DI VIDEOSORVEGLIANZA

Tesi di laurea di:

Riccardo Carlesso

Relatore:

Prof. Ing. **Luigi Di Stefano**

Correlatori:

Prof. **Massimo Ferri**

Ing. **Martino Mola**

Sessione Straordinaria

Anno Accademico 2000-2001

Parole chiave:

Matching
Texture Analysis
Tracking
Videosorveglianza
Visione artificiale

Argenta, 12 marzo 2002

Alla mia Crysania, per avermi fatto dismettere le Vesti Nere
in favore delle Rosse
Alla mia famiglia per la pazienza e la disponibilità
A Luigi e Martino per lo stesso motivo
A Ridge per essersi laureato il 20 Marzo solo per farmi compagnia
Al Cuore e all'Alma Mater
A Massimo per avermi insegnato che insegnare non è poi così male
Al LaTeX che ha reso divertente persino scrivere la tesi

There is a theory which states that if ever anyone discovers exactly what the Universe is for and why it is here, it will instantly disappear and be replaced by something even more bizarre and inexplicable.

There is another theory which states that this has already happened.

(Douglas Adams)

Indice

Capitolo 1

Introduzione

> The beginning of the knowledge is the discovery of something we do not understand.
>
> *Frank Herbert*

1.1 Visione artificiale e Tracking

L'obiettivo di questa tesi è quello di sperimentare la capacità dei cosiddetti *appearance models* nel caratterizzare immagini in movimento con lo scopo di risolvere problemi di tracking in un sistema di videosorveglianza.

Lo scopo dei sistemi di videosorveglianza è quello di identificare oggetti, persone, veicoli, o di riconoscere situazioni di interesse in maniera automatica a partire da filmati raccolti da telecamere. Queste telecamere forniscono una sequenza di immagini digitalizzate a un computer il cui processo di analisi dell'immagine può essere schematizzato in due stadi: dapprima vengono analizzate da un algoritmo di basso livello, nel secondo da un modulo di tracking.

1. Il compito di un algoritmo di basso livello è quello di analizzare un'immagine della sequenza e decidere, per ogni pixel, se questo appartiene a un oggetto di interesse o allo sfondo; questa classificazione però è relativa a una singola immagine, e a questo livello di elaborazione non viene data nessuna informazione sulle relazioni esistenti tra gruppi di pixel appartenenti a immagini consecutive. Tutto ciò che si ha sono agglomerati di pixel che si scostano 'troppo' da quello che è il background.

2. A partire da questo tipo di informazione il modulo di tracking deve essere in grado di accorgersi dell'apparizione di un oggetto e inseguirlo per tutta la durata della scena fino alla sua scomparsa. Per svolgere questo compito principale il sistema dovrà essere in grado di risolvere diversi problemi tra i quali il riconoscimento e la correzione degli errori fatti dal basso livello e la comprensione di quei casi in cui vari oggetti si sovrappongono per un certo intervallo di tempo. Sia chiaro che questo modulo non è lo stadio finale del sistema, ma un intermediario che deve fornire funzionalità ad un ipotetico modulo di alto livello *in grado di interpretare la scena*, che può essere un ulteriore strato software o l'utente finale. Un apparato di tracking si propone auspicabilmente di risolvere una vasta classe di problemi *senza introdurre una semantica della scena osservata*: questo con lo scopo di avere uno strato software quanto più possibile *modulare* e applicabile ad un ampia casistica (sicurezza domestica, monitoraggio stradale, rilevazioni outdoor, ...).

Il lavoro di ricerca di questa tesi è stato svolto presso il laboratorio di Visione Artificiale del DEIS (Dipartimento di Elettronica, Informatica e Sistemistica), della Facoltà di Ingegneria dell'Università di Bologna. Nello stesso laboratorio sono stati svolti in passato studi sugli algoritmi di basso livello per la rilevazione di oggetti di interesse in sequenze video [18]; il sistema sviluppato in questa tesi dovrà ricevere in ingresso i risultati prodotti dal migliore di questi algoritmi e fornirà in uscita un'interpretazione della scena basata sul tracking.

Questo sistema dovrà inoltre essere il più possibile generale nel senso che non verranno fatte ipotesi sulla natura degli oggetti protagonisti della scena, dovrà identificare allo stesso modo persone, auto o qualsiasi altra cosa. Ciò che si deve affrontare è un tipico problema che viene risolto senza particolari problemi da una mente umana, mentre (allo stato dell'arte) nessun algoritmo è in grado di affrontarlo con prestazioni simili a quelle umane. La ragione di ciò sta nel fatto che per risolvere il problema, l'uomo usa un'insieme di ragionamenti e di informazioni a diversi livelli (non solo visivo) che in pratica non è riproducibile da una macchina.

1.2 Obiettivi di questa tesi

Lo scopo di questa tesi è di integrare un sistema di tracking (già presente e funzionante) allo scopo di superare i suoi limiti di *riconoscimento* di immagini. Un modulo di tracking si deve infatti occupare di problemi molto diversi e in buona misura *può* non preoccuparsi di riconoscere gli oggetti (attraverso la loro forma, i colori, e altre caratteristiche) che insegue. Nella maggior parte del proprio lavoro, l'inseguimento dell'oggetto è un compito facile, poiché l'oggetto si muove lentamente e si possono collegare tra loro istanze successive dello stesso oggetto con molto agio. Vi sono però casi in cui due (o più) oggetti si intersecano, tornando eventualmente a separarsi in un secondo momento, oppure questi possono nascere e morire (entrando o uscendo da una porta o – peggio ancora – uscendo da un altro oggetto, come per esempio un'automobile).

Vi sono casi semplici (per un essere umano) in grado di mettere in crisi un qualunque sistema di tracking.

Poichè il suo compito è arduo, si stanno sviluppando modelli non completamente deterministici in cui si cerca di trovare solo quelle associazioni certe lasciando in sospeso relazioni ignote (nella speranza che in un secondo momento si possa risolvere il problema). Questo è stato il metodo scelto dal nostro gruppo di visione: si cerca di prendere decisioni solo quando queste siano particolarmente sicure, portando innegabilmente due vantaggi:

1. in un secondo momento si potrebbe raggiungere una maggiore certezza sulla scelta da fare;

2. un ipotetico modulo decisore (sempre interno al tracking) potrebbe applicare un tipo di conoscenza più pregiata alle relazioni lasciate in sospeso, per esempio studiando il grafo degli oggetti di cui si è persa traccia; si possono inferire notevoli informazioni da una conoscenza sul problema che duri per un lungo tempo. Questa è *effettivamente* una direzione presa dal gruppo di lavoro al momento.

Molti problemi possono comunque rientrare in un caso semplice da descrivere: la *mutua occlusione* tra più oggetti. Se si guarda il problema con gli occhi di una

telecamera, in questo caso vi sono più macchie che si mescolano e poi tornano ad essere delle entità separate (per esempio, anche solo due persone che si stringono la mano).

Un modulo di tracking deve essere in grado di stabilire una relazione tra gli oggetti che esistevano prima della fusione (*merge*) e quelli che sono nati dopo.

É proprio qui che s'inserirà il sistema studiato in questa tesi. Esso si propone fondamentalmente di:

- caratterizzare le singole *immagini* associate agli oggetti, attraverso l'estrazione automatica di features di forma e di colore;
- descrivere coerentemente in modo robusto *sequenze* di immagini, basandosi sul fatto che certi descrittori saranno più propensi a *caratterizzare* certe immagini che non altri, e questo potrà variare da caso a caso;
- definire delle metriche di confronto tra immagini basate sulle features estratte in precedenza e usate in caso vi sia una qualche rilevante relazione tra più oggetti;
- utilizzare queste metriche per stabilire una *Matrice di Matching* che sia il più possibile in grado di risolvere relazioni pre- e post-occlusione (decidendo anche se stabilire o meno una relazione, in attesa di dati più sicuri).

La Visione Artificiale fa uso notevole di *descrittori* per la caratterizzazione di immagini già da tre decenni; vi è una grossa differenza, però, tra l'elaborazione di un'immagine e quella di una sequenza; è richiesta una potenza di calcolo notevole e quindi in generale lo strumento mal si presta al caso di sequenze animate. D'altro canto, l'aumentare delle prestazioni dei calcolatori e il fatto che le immagini (corrispondenti agli *oggetti*) che vengono restituite dal modulo di basso livello siano tipicamente molto piccole (quindi agevoli da calcolare) ha giustificato questo approccio in questo tipo di contesto.

L'idea fondamentale e pervasiva di tutta la tesi è che molti descrittori possano concorrere a definire non solo un'immagine, ma anche una sequenza, e che anzi il loro potere di analisi abbia a guadagnarne notevolmente: calcolando gli stessi parametri su istanze successive di un oggetto inseguito (molto spesso un oggetto viene agevolmente inseguito per molte decine di frame prima di essere

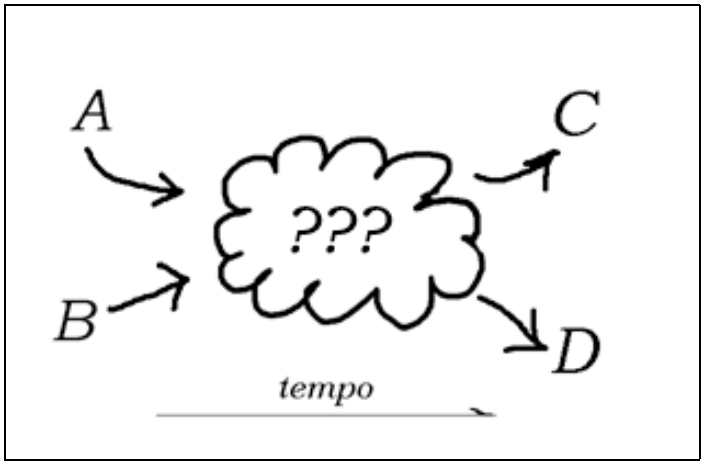

Figura 1.1: Semplice esempio di collisione.

coinvolto in un merge) si può costruire una *statistica* e ci si può accorgere in maniera automatica di quali descrittori siano meglio in grado di *catturare l'idea* dell'oggetto stesso. Il sistema sarà allora in grado, all'occorrenza, di riconoscere un oggetto basandosi proprio su quelle feature che si erano precedentemente rivelate più stabili. Si potrà allora catturare bene l'idea di automobile attraverso la sua forma, mentre lo stesso descrittore fallirà per corpi non rigidi come una persona (per cui magari vi saranno altre 'invarianti'); nel caso vi sia un split si potrà allora stimare la probabilità che un oggetto appena nato (non classificato) sia l'evoluzione o meno di uno degli oggetti coinvolti nella fusione.

Sono stati dunque studiati e provati molti descrittori non tanto con lo scopo di trovare la rosa dei più adatti a descrivere immagini (poiché questa rosa dipenderebbe inevitabilmente dal contesto del problema), quanto per costruire un gruppo che fornisse un buon compromesso qualità e quantità vs tempo di calcolo. Sono state usate feature di forma (perlopiù i Momenti), di colore (istogrammi dei toni di grigio e qualche valore locale) e di tessitura (informazioni ancora più pregiate sul colore che ne definiscono la correlazione, con valori come contrasto, omogeneità, entropia, rugosità, ...).

Purtroppo, un sistema reale è *aperto* e deve prevedere la nascita e la morte di oggetti.

Il sistema dovrà dunque essere in grado di dire sia *"l'oggetto B non è altro che l'oggetto A"* sia *"l'oggetto B non assomiglia sufficientemente a nessuno quindi*

diremo che è appena nato". Questo pone notevoli problemi poiché impone un'informazione molto pregiata dell'ambiente da misurare: è molto più facile (vedere fig. 1.1) accoppiare due coppie ($A, B \rightarrow C, D$) che non ammettere il caso in cui vi sia una nascita e una morte, poiché non è più sufficiente uno studio comparato (in cui è sufficiente che il sistema sbagli *meno* per la coppia vera che per la coppia falsa) ma uno studio assoluto: "A è C?", "A è D?" e così via.

1.3 Strumenti usati

Come detto in precedenza il sistema che si deve realizzare dovrà interpretare i risultati forniti dagli algoritmi di basso livello già studiati e sviluppati in precedenti lavori: in particolare è già stato implementato un sistema software (il *"Sentinel"*), usando il linguaggio C++, in grado di acquisire sequenze di immagini ed elaborarle usando i suddetti algoritmi.

La scelta di un linguaggio di programmazione come il C++ è stata dettata principalmente dalla semplicità d'*integrazione* (passata e futura) con l'algoritmo di basso livello (scritto da A. Fabbri, [18]) in questo linguaggio. Quella scelta era stata a sua volta guidata dall'efficienza di un linguaggio come il C associata alla modularità e manutenibilità che solo un paradigma a oggetti oggi può offrire. Sono state create classi che facciano da *servitore* al modulo di tracking (un sistema esperto a regole progettato da M. Mola, [17]), con la possibilità di utilizzare insiemi leggeri o pesanti di descrittori (secondo l'esigenza).

Per motivi contingenti il sistema è stato testato solo su sequenze off-line ma nell'ottica di una futura integrazione col sistema di tracking; questa è un'ulteriore giustificazione della scelta di un'architettura orientata agli oggetti. Nella Sez. 5.5 vi sono stime di costo per un'integrazione in un sistema di tracking online.

Si è dunque lavorato su sequenze di bitmap in memoria di massa estratte precedentemente da un algoritmo di *foreground extraction*.

1.4 Guida alla lettura

Il capitolo 2 presenterà teorie e basi matematiche necessarie alla definizione delle *features* e alla realtà che vogliono descrivere.[1]

Il capitolo 3 presenterà invece i modelli usati per *associare* questi *descrittori* a immagini (e successioni di queste nel tempo) necessari a definire opportune *metriche* su questi oggetti.

Il capitolo 4 offrirà una panoramica sulle *metriche* usate per confrontare tra loro più immagini e più sequenze.

Il capitolo 5 descriverà i *risultati sperimentali* cui hanno portato questi modelli.

Il capitolo 6, infine, raccoglierà le conclusioni del lavoro svolto.

[1] quindi anche modelli generali per le immagini.

Capitolo 2

Descrittori per la caratterizzazione di immagini

> Bene, ma rifletti; non abbiamo convenuto più volte che nomi dati con giudizio sono immagine e somiglianza della cosa a cui danno il nome?
>
> *Socrate*

La letteratura specifica presenta due grossi filoni per rappresentare e soprattutto confrontare immagini: il *"template matching"* e gli *"appearance models"* (o *"features extraction"*).

Il primo consiste nel ricercare un certo *pattern* noto a priori in un'immagine ignota. É un'operazione molto interessante, utile perlopiù a risolvere una classe molto specifica di problemi (i.e. trovare il numero di chiodi in un piano di lavoro, dire se è presente una persona – o meglio una testa – in un'immagine).

Un approccio certamente più generale consiste nell'estrarre da una generica immagine parametri che si ritengono essere il più possibile *rappresentativi* di essa; che siano quindi ragionevolmente robusti al rumore ma che discriminino fortemente immagini 'diverse', secondo una metrica simile a quella dell'occhio umano.

Nella presente tesi si è usato il secondo metodo in quanto più più generico e atto a risolvere un maggior numero di problemi *senza conoscenza pregressa.*

Questo capitolo tenta di descrivere tutto l'insieme dei descrittori che sono stati effettivamente usati in questo lavoro. Essendo questi in gran numero, sono

stati divisi in una gerarchia di famiglie e sottofamiglie in modo da comprenderne meglio il quadro complessivo.

2.1 Introduzione

Da questo momento in poi verranno dati per scontati certi strumenti e terminologie; questa sezione ha dunque lo scopo di riunire quanto più possibile le conoscenze propedeutiche al capitolo.

Per **bitmap** s'intende un'immagine rettangolare (una *mappa di bit*, appunto, o una matrice) che associa a ogni punto una tonalità di grigio (si assumono di base 256 possibili tonalità di grigio, e verrà associato a questa cardinalità il simbolo G). Tutta la tesi è stata svolta su immagini a livelli di grigio , dunque non ha senso modellare l'immagine a più canali di colore. Qualora il fattore colore sia particolarmente interessante, verrà specificato in seguito.

Si farà spesso riferimento al termine **Frame**; questo è l'immagine bitmap catturata dal *frame-grabber* che è supposta a **sfondo fisso** e che contiene tutti gli oggetti in movimento che i moduli di basso livello devono restituire ai livelli superiori del 'motore' d'elaborazione. In questo lavoro non si è mai avuto a che fare con il frame nella sua interezza ma solo con immagini già *segmentate*, quindi **sviscerate dal contesto**. Vi sono pesanti implicazioni: non si possono per esempio fare considerazioni *a priori* sull'istante d'ingresso di un oggetto rispetto a un altro, né conoscere la posizione del baricentro dell'oggetto *all'interno* del frame.

Quando si parla di **immagine** s'intende una coppia di bitmap originale-maschera, che nel loro insieme determinano univocamente l'immagine stessa.

1. L'*originale* consiste di un rettangolo a toni di grigio che contiene tutta l'immagine e (spesso) pixel appartenenti allo sfondo.

2. La *maschera* (detta anche BLOB)[1] consiste invece di un rettangolo bitonale (ogni pixel *o* è bianco *o* è nero) che permette di discriminare tra i pixel oggetto e i pixel sfondo.

[1] detta anche **BLOB**, acronimo per Binary Large OBject. Questa sigla trova applicazione in vari campi (database, crittografia, ...), qui significa: 'macchia binaria'.

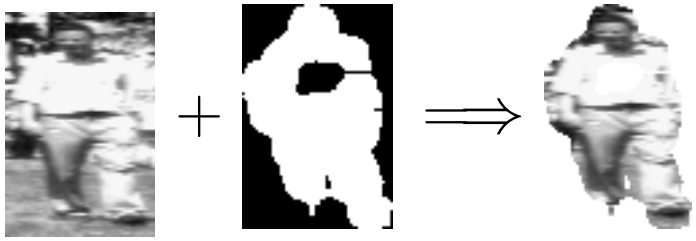

Figura 2.1: rappresentazione di un'immagine: originale, maschera e risultato finale.

Una definizione formale può essere:

Definizione 2.1.1 *Sia $\mathcal{I}[X, Y, G]$ un'immagine digitale $\mathcal{I}$ di dimensioni $(X \times Y)$ a G possibili toni di grigio. Rappresenteremo l'immagine con la coppia $(\mathcal{D}, g)$, dove $\mathcal{D}$ è il dominio di definizione $\mathcal{D} \subseteq (\mathbb{N}_X, \mathbb{N}_Y)^2$; g è invece la funzione che mappa ogni punto della matrice di punti (di dimensione $X \times Y$) in un possibile tono di grigio: $g : \mathcal{D} \subseteq (X, Y) \longrightarrow \mathbb{N}^*_{G-1}$[3], dove $g(x, y)$ non è altro che il tono di grigio assunto dall'immagine nel punto (x, y). Definisco inoltre $\mathcal{S}_{\mathcal{I}_G}$ lo spazio di tutte le possibili immagini a G toni di grigio.*

Questa astrusa definizione ben si sposa con il supporto fisico delle immagini su cui si è lavorato: $\mathcal{D}$ non è altri che la *maschera*, mentre $g(x, y)$ è l'*originale. Da notare che per gli scopi della trattazione* $\mathbf{G}$ *rimarrà sempre costante e quindi, spesso, verrà ignorato.*

In fig. 2.1 si può riscontrare la terminologia adottata: a sinistra c'è l'originale (porzione rettangolare del frame iniziale); in mezzo la maschera (con in bianco i pixel che appartengono all'oggetto); a destra, infine, si può osservare il risultato in cui si sono colorati (artificialmente) di bianco *anche* i pixel dello sfondo. Si noti che l'informazione della maschera è particolarmente sporca; ciò si deve al fatto che in quell'istante l'algoritmo di estrazione non ha fornito l'output che un essere umano avrebbe ipoteticamente prodotto; questo influisce negativamente sull'elaborazione che avverrà in seguito. Si noti infine che il blob contiene una vistosa falla al centro; questo è dovuto al fatto che parte della maglietta dell'oggetto viene confusa con lo sfondo.

[2]Con $\mathbb{N}_M$ s'intende l'insieme dei numeri naturali minori o uguali a M.

[3]si definisce invece $\mathbb{N}^0_L$ (o $\mathbb{N}^*_L$) l'insieme dei numeri naturali minori o uguali a L, compreso lo zero.

Ricordiamo infine che un'immagine è comunque costituita di pixel e semplici operazioni geometriche (rotazione, variazione di scala, ...) possono distruggere o comunque compromettere l'immagine originale, soprattutto se la sua definizione non è superiore a quella necessaria al suo studio. Ogni operazione su un'immagine impone di rimettere in discussione ogni pigmento di colore (o di grigio) e quindi può richiedere un ricalcolo (approssimato) che può essere visto come un ulteriore *rumore di quantizzazione*.

Si parlerà, talvolta, di **pixel oggetto** intendendo tutti e soli quei pixel che *appartengano* al supporto dell'immagine.

2.2 Proprietà dei descrittori

Qui di seguito vengono elencate e descritte le regole con cui si sono cercate di descrivere sequenze di immagini durante il presente lavoro. Un descrittore vuole catturare un particolare dell'immagine che può essere l'area, il perimetro, il colore medio, ... Vediamo però come dovrebbe comportarsi un descrittore *ideale* per poi vedere con quale metro giudicare quelli effettivamente usati. Un descrittore è *rappresentativo* per un'immagine se è *stabile nel tempo per essa e se distingue bene tra due immagini diverse*.

La bontà di un descrittore non può essere svincolata dal contesto: in certi casi alcuni falliscono mentre altri catturano bene l'idea dell'immagine. Occorre quindi decidere il problema da risolvere (classificazione di immagini, tracking di sequenze, ...) e le immagini che vi compaiono (persone, foglie, mammografie, automobili...)

Non ha senso un giudizio assoluto di un descrittore, poiché nessuno a priori è buono o cattivo finché non si conosce il problema (discriminare persone tra loro, foglie diverse, tumori benigni dai maligni, ...).

2.2.1 Invarianza dei descrittori

Certi semplici descrittori catturano molto bene l'idea di un'immagine (per esempio l'area), ma occorre riflettere sul fatto che questi possano variare nel tempo ro-

vinando l'informazione. Fondamentalmente ci sono due cause che fanno cambiare il valore di un descrittore: il rumore e il movimento stesso dell'immagine.

Entrambe questa cause sono inevitabili, però sulla seconda sono stati fatti molti studi e si è giunti a premiare descrittori che rimangano particolarmente stabili pur cambiando l'immagine. Un oggetto inquadrato da telecamera fissa può avvicinarsi ad essa (cambiando le proprie dimensioni), può ruotare o girarsi. In ciascuno di questi casi l'immagine dal punto di vista del computer viene stravolta.

Ecco perché diventa necessario individuare dei descrittori che siano immuni o comunque robusti di fronte a queste variazioni. Parleremo allora di *invarianza* rispetto a un certo fattore. Segue un elenco delle più comuni trasformazioni:

Rotazione. Molto spesso l'invarianza per rotazione deve essere raggiunta. In casi in cui la posizione sia importante (es: OCR) questo non è più vero. La rotazione di un'immagine digitale è il più possibile evitata *non solo* per il costo computazionale (che, a seconda dell'applicazione, può essere tranquillamente sopportato) quanto per l'errore di quantizzazione introdotto – *l'immagine va infatti ricalcolata poiché i pixel non corrispondono più.* Ecco perché è impensabile ad esempio estrarre *features* di tessitura in un'immagine ruotata: l'immagine risultante sarebbe così diversa dall'originale da non esserne più rappresentativa.

Traslazione. Se l'oggetto si muove semplicemente traslando, la maggior parte dei suoi descrittori non cambierà valore. A meno che non sia specificato, tutti i descrittori trattati sono invarianti per traslazione.

Scala. Questo fattore è il più distruttivo di tutti poiché altera completamente le caratteristiche dell'immagine. Non solo cambia le dimensioni e quindi tutti i descrittori più semplici ma anche la tessitura (da lontano un oggetto può apparire uniforme mentre da vicino può apparire ricco di variazioni). Inoltre, quand'anche si riesca a trovare un descrittore invariante per scala, a causa della *quantizzazione* dell'immagine, ogni formula risulterà indubbiamente molto disturbata da questa variazione. L'unica famiglia che non sembra risentire da questo problema sembra essere quella delle statistiche del prim'ordine (gl'istogrammi di colore).

2.2.2 Normalizzazione dei descrittori.

Per il 95% dei descrittori è stata scritta la versione standard e la versione normalizzata. La seconda non è altro che la moltiplicazione del primo numero per un opportuno coefficiente che la porti da 0 a 1. Questo non può essere fatto in due casi: se la funzione *non* ammette fondoscala (il che in campo informatico accade difficilmente), o se la sua normalizzazione le toglie il significato originale. Ciò è successo ad esempio per l'area dell'oggetto: la sua normalizzazione più ovvia è la divisione per l'area della bounding box; il descrittore area però non dirà più quanto un oggetto è *grande* ma piuttosto quanto copre l'area assegnatagli. Questo chiaramente compromette la semantica del descrittore (tant'è che, in effetti, esiste un altro descrittore calcolato proprio così). La divisione, d'altro canto, per una costante sufficientemente grande andrebbe benissimo ma, in casi pratici, altererebbe la statistica dell'oggetto (avendo varianze più basse di altri oggetti con lo stesso dominio).

Sono rimasti, dunque, anche descrittori non normalizzati; per un loro studio comparato, è stata adottata una strategia semplice ed efficace.

Poiché questi descrittori verranno inglobati in una funzione distanza, occorrerà definire un'interfaccia comune, quindi fondamentalmente un dominio di definizione. L'unica ipotesi che è stata fatta è che i valori siano sempre *non negativi* (questo ha talvolta forzato dei risultati al proprio valore assoluto). Possono quindi esserci descrittori che variano da 0 a 1 (un vincolo di normalizzazione che è stato *imposto* quando possibile) e altri che invece sono dell'ordine di grandezza di milioni. Se si conoscesse *a priori* il campo di variabilità delle variabili aleatorie il problema non si porrebbe: basterebbe confrontare tra loro le versioni normalizzate. E se questo non è possibile? La soluzione adottata, sebbene formalmente scorretta, pare funzionare molto bene: si confrontano le statistiche del secondo ordine (fondamentalmente le varianze) normalizzandole per il valore medio. Questo a regime dà risultati sperimentali buoni, mentre dà risultati sballati solo nei primi campioni (il tempo di convergenza è peraltro buono, dell'ordine di pochi frame).

Per persuadersi del fatto che l'approccio è formalmente sbagliato, basti pensare a due variabili aleatorie gaussiane (interne al range $[0, 1]$) con la stessa σ

(diciamo 0,01) ma media diversa ($0,2$ e $0,8$ rispettivamente). Potrebbe essere il colore normalizzato del maglione di due persone che cambia molto poco nel tempo, l'uno chiaro l'altro scuro. Con l'approccio teorico (che in questo caso *è* possibile) le due variabili verrebbero ritenute *altrettanto* stabili (il che è giusto). Con questo metodo, invece, si trovano due $\bar{\sigma} := \frac{\sigma}{\mu}$ diverse (0,05 e $0,0125$ rispettivamente) portando alla conclusione che il secondo descrittore è *4 volte più stabile*.

Insomma, in teoria la normalizzazione apportata non è corretta, ma in pratica funziona bene (ai fini dello studio, una variabile come la seconda è *molto* più stabile poiché la prima naviga troppo vicina allo zero).

2.3 Descrittori semplici

In questa famiglia ricadono tutti quei descrittori che per la loro semplicità non richiedono alcun tipo di elaborazione dell'immagine[4] . Questi sono:

1. Descrittori sulla *Bounding box* (il più piccolo rettangolo contenente l'immagine)[5]:

   - h_{bb}. L'altezza del più piccolo rettangolo contenente l'immagine.
   - w_{bb}. La larghezza del rettangolo.
   - A_{bb}. L'area del rettangolo.
   - λ_{bb}, Elongazione verticale. Rapporto tra h_{bb} e w_{bb}. É un ottimo descrittore per distinguere persone da veicoli (ammesso che non sia loro concesso di ruotare). Da non confondere con l'elongazione, che in letteratura è il rapporto tra l'asse maggiore e l'asse minore d'inerzia. É invariante per scala.

[4] Ad esser più precisi dovremmo dire che non richiedono alcun *ciclo* per essere calcolati

[5] A dir la verità, nell'implementazione attuale la Bounding Box è fatta in modo che in ogni frame di una sequenza animata due di queste non si possano intersecare. Se due oggetti sono sufficientemente vicini, la loro bounding box sarà *per entrambi* il più piccolo rettangolo che li contiene *tutti e due*, per esempio. Quest'ulteriore vincolo la defferenzia dal *Minimum Enclosing Rectangle* (MER) della letteratura.

2. Descrittori locali di colore:

   - G_c, GRIGIO CENTRALE. Il valore di grigio del pixel centrale della bounding box di coordinate $(\frac{x}{2}, \frac{y}{2})$ (che non necessariamente è il baricentro e può addirittura non appartenere alla maschera). Poiché questo valore era troppo rumoroso, è stato fatto uno smoothing con gli 8 pixel vicini. Nel caso di una persona questo tende ad essere all'altezza del cuore
   - G_h, GRIGIO 'ALTO'. Il valore di grigio, anch'esso mediato, del punto $(\frac{x}{2}, \frac{3y}{4})$. Nel caso di persone questo si trova circa all'altezza della testa.
   - G_l, GRIGIO 'BASSO'. Il valore di grigio, anch'esso mediato, del punto $(\frac{x}{2}, \frac{3y}{4})$. Per persone questo *può* indicare il colore dei pantaloni (gonna, ...).

2.4 Descrittori di maschera

In questa famiglia ricadono i descrittori per calcolare i quali è sufficiente la maschera (ovvero la bitmap a due toni che ci dice semplicemente quali pixel appartengono all'oggetto e quali no). Vi afferiscono tutte le features **di contorno** (poiché la nozione di contorno è presente solo in questa immagine) e i **Momenti**.

2.4.1 Momenti

I momenti sono un'intera famiglia di caratteristiche di un'immagine[6], e nel loro insieme riescono a determinarla univocamente (come dimostrato in [7]). Si è però scoperto che un piccolo sottoinsieme è particolarmente significativo per descrivere l'immagine stessa.

Sia data un'immagine digitale I di dimensioni $(X \times Y)$ a G possibili toni di grigio (adotteremo il range 'informatico': 0..G-1). Si può dunque modellare

[6] I momenti sono uno strumento potentissimo usato in ogni campo scientifico in cui si abbia a che fare con una distribuzione spaziale di un oggetto con proprietà locali quali la densità di massa (in meccanica), i toni di grigio (nel nostro caso), la densità di carica, ... Sono in generale un potente strumento d'ispezione per catturare features globali di un oggetto definito in maniera locale.

l'immagine con una funzione $g : \mathcal{D} \subseteq (X, Y) \longrightarrow \mathbb{N}^*_{G-1}$, dove $g(x, y)$ non è altro che il tono di grigio assunto dall'immagine nel punto (x, y).

Vediamo di definire genericamente tutti i momenti possibili, per poi focalizzare l'attenzione su quelli più interessanti. Definiamo dunque i momenti spaziali M_{ij} e i momenti centrali μ_{ij}:

$$\mathbf{M_{ij}} = \sum_{\substack{x \\ (x,y)\in\mathcal{D}}} \sum_{y} x^i \, y^j \, g(x, y) \tag{2.1}$$

$$\mu_{\mathbf{ij}} = \sum_{\substack{x \\ (x,y)\in\mathcal{D}}} \sum_{y} (x - G_x)^i \cdot (y - G_y)^j \cdot g(x, y), \tag{2.2}$$

dove:

$$\mathbf{G} \equiv (G_x, G_y) = \left(\frac{M_{10}}{M_{00}}, \frac{M_{01}}{M_{00}} \right) \tag{2.3}$$

Infine sono interessanti i momenti centrali normalizzati $\eta_{\mathbf{ij}}$:

$$\eta_{\mathbf{ij}} = \frac{\mu_{ij}}{\mu_{00}^{\gamma}}, \quad \text{con } \gamma = \frac{i + j}{2} + 1$$

Viene definito *ordine* di un momento la somma dei due indici: $\gamma = i + j$.

Queste formule possono essere applicate sia a un'immagine a toni di grigio che a un'immagine bitonale (nel nostro caso, alla maschera), con un significato leggermente diverso. Nel secondo caso ci si disinteressa del fattore *colore* in favore della sola *forma* dell'oggetto (poiché il contributo non dipende più dal tono di grigio ma solo dal fatto che il punto appartenga o meno all'oggetto).

A queste tre famiglie di momenti può essere attribuito un semplice significato:

- I momenti M_{ij} possono esser visti come una media dei valori di grigio $g(x, y)$ ponderata da un peso che è funzione della posizione. Per un'immagine bitonale, i momenti danno:
  - ordine 0: la *massa* (o l'*Area* se è uniforme);
  - ordine 1: il *Baricentro* (vedi eq. (2.3), pag. 17):

– ordine 2: la Matrice d'Inerzia.

$$I = \begin{pmatrix} M_{20} & M_{11} \\ M_{11} & M_{02} \end{pmatrix} \tag{2.4}$$

In particolare gli autovalori (λ_1, λ_2) ci forniscono la lunghezza degli *assi principali d'inerzia*, e gli autovettori $(\mathbf{v}_1, \mathbf{v}_2)$ le rispettive direzioni.

Quest'analogia è possibile poiché le definizioni meccaniche di *massa*, *baricentro* e *momenti d'inerzia* usano le stesse formule, sostituendo ai toni di grigio la densità di massa (quindi possiamo pensare all'uso della maschera come a un'ipotesi di densità *uniforme* del corpo stesso).

L'insieme di tutti i possibili momenti definisce univocamente l'immagine ma un ottimo *trade-off* tra calcolo ed effettivo significato fisico è considerato il livello 3 (vedi Hu, Sez. 2.4.2). Nel caso di un'immagine non bitonale il discorso vale ancora se pensiamo alla scala di grigio come a una densità di massa.

- I momenti μ_{ij} non sono altro che una variante dei momenti spaziali in cui il sistema di riferimento è baricentrale. Questi diventano dunque *invarianti per traslazione*.

- I momenti η_{ij} sono una normalizzazione dei momenti centrali che li rende *invarianti per scala*.

- Sarebbe possibile rendere i momenti *invarianti per rotazione* cambiando il sistema di riferimento nel sistema inerziale avente per base gli autovettori della matrice d'inerzia. Sebbene concettualmente vero, spesso non ha senso ricampionare l'immagine nel nuovo sistema e ci si accontenta di estrarre alcune *features* basate su queste informazioni.

2.4.2 Gl'invarianti di Hu

Hu ha dimostrato nel 1962 [13] che esistono 7 features invarianti per scala, rotazione e traslazione che possono essere costruiti a partire dai momenti centrali normalizzati (η_{ij}) di ordine fino al 3. Esse sono $[\phi_1, \phi_2, ..., \phi_7]$:

$$\phi_1 = \eta_{20} + \eta_{02} \tag{2.5}$$

$$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \tag{2.6}$$

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (\eta_{03} - 3\eta_{21})^2 \tag{2.7}$$

$$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{03} + \eta_{21})^2 \tag{2.8}$$

$$\begin{aligned}\phi_5 = {} & (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + \\ & +(\eta_{03} - 3\eta_{21})(\eta_{03} + \eta_{21})[(\eta_{03} + \eta_{21})^2 - 3(\eta_{12} + \eta_{30})^2]\end{aligned} \tag{2.9}$$

$$\begin{aligned}\phi_6 = {} & (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + \\ & +4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{03} + \eta_{21})\end{aligned} \tag{2.10}$$

$$\begin{aligned}\phi_7 = {} & (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + \\ & +(3\eta_{12} - \eta_{30})(\eta_{03} + \eta_{21})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]\end{aligned} \tag{2.11}$$

Si è riscontrato che i valori più stabili e rappresentativi tra i sette invarianti sono i primi 3-4. Sono stati dunque tagliati dalle metriche i parametri (ϕ_5, ϕ_6, ϕ_7).

2.4.3 Contorni

Una forte scelta progettuale del sistema è stata quella di **trascurare il più possibile informazioni sul contorno**. Osservando infatti le immagini del *data-set* sono state trovate istanze successive (nel tempo) di uno stesso oggetto molto variabili come forma. Il motivi da ricercarsi sono fondamentalmente due:

1. Le applicazioni di Tracking tipicamente richiedono di poter lavorare su oggetti deformabili; le sagome delle persone, per esempio, hanno contorni *intrinsecamente* molto variabili (basti pensare al movimento di gambe e braccia in una persona vista di profilo durante una camminata).

2. Il modulo di basso livello atto a estrarre il *foreground* dal background: se certi pixel sono visti simili al background, ci troviamo con un'immagine cui mancano dei punti; per un motivo analogo, l'immagine può presentare pixel che non sono suoi ma appartengono allo sfondo. Ci si può accorgere del fatto che tutti i descrittori *di area* risentono molto meno dell'effetto di questo 'rumore di elaborazione' rispetto a features 'di contorno'. Proprio per

questo i secondi sono stati ritenuti meno atti a rappresentare un'immagine rispetto ai primi. Sono dunque stati estratti solo i più semplici e famosi in letteratura evitando di sfruttarli pesantemente come avrebbe permesso una diversa situazione per i dati in input.

C'è da dire, tuttavia, che il sistema prevede di studiare non solo il valore di un descrittore, ma anche la sua *variabilità*. In quest'ottica, i descrittori di contorno potrebbero aiutare a discriminare tra corpi *rigidi* e *deformabili*.

Vediamo i descrittori che sono stati usati:

p, Perimetro dell'oggetto [7]: è la lunghezza del bordo dell'immagine, ovvero il numero di pixel appartenenti all'immagine e a distanza unitaria (secondo una metrica - le più usate sono la *8-neighborhood* e la *4-neighborhood*) da un pixel dello sfondo.

$\mathcal{C}$, Compattezza Essa è definita come:

$$\mathcal{C} = \frac{4\pi\mathcal{A}}{p^2}, \tag{2.12}$$

dove p è il perimetro poc'anzi citato e $\mathcal{A}$ l'area della maschera.

Questa famosissima feature gode di notevoli proprietà: è invariante per rotazione e scala, varia da 0 a 1, e vale 1 solo nel caso di un cerchio perfetto. É molto stabile. É probabilmente il più semplice ed elegante (nel senso di conciso, pregnante) descrittore di *forma*. Purtroppo nel presente sistema, come già detto, il bordo stesso è un input fortemente degradato.

2.5 Descrittori sull'immagine a toni di grigio

In questa sezione rientrano tutti quei descrittori che sfruttano l'intera informazione associata all'immagine: la matrice dei valori in *gray-scale* e la maschera stessa.

[7] In effetti non è garantita a priori la connessione completa dell'oggetto, quindi questi può avere più perimetri. La libreria grafica CV [1] restituisce una lista di contorni e i calcoli vengono effettuati sul *primo*.

2.5.1 Statistiche di grigio del prim'ordine (istogramma)

Per statistiche di colore **del prim'ordine** s'intendono quei calcoli sul tono di grigio che non dipendono dalla posizione *reciproca* e assoluta dei pixel.

L'unica statistica del genere trovata in letteratura è l'istogramma dei toni di grigio. Sia I l'immagine digitale a G toni di grigio e $g : \mathcal{D} \subseteq (\mathbb{N}_X, \mathbb{N}_Y) \longrightarrow \mathbb{N}^0_{G-1}$ la funzione che mappa l'insieme dei pixel nei possibili gray-level. Definiamo allora $\mathcal{H} \equiv (\mathcal{H}_0, ..., \mathcal{H}_{G-1})$, dove:

$$\mathcal{H}_k = Card\{(x, y) \in \mathcal{D} | g(x, y) = k\} \tag{2.13}$$

Poiché spesso la struttura dati contenente l'istogramma è grossa, dipendentemente dall'uso che se ne deve fare si può sentire l'esigenza di ridurre la bontà della misura aumentando la *grana* dell'istogramma (quindi riducendo la *granularità* della misura). Si parla allora di suddividere l'istogramma in un certo numero N di *bins* grossi esattamente $\xi \doteq \frac{G}{N}$ e di computare dunque un istogramma ridotto $\mathcal{H}' \equiv (\mathcal{H}'_0, ..., \mathcal{H}'_{N-1})$, dove:

$$\mathcal{H}'_{k,Dim} = Card\{(x, y) \in \mathcal{D} \mid k \cdot \xi \leq g(x, y) < (k + 1)\xi\} \tag{2.14}$$

Senza perdere in generalità, d'ora in poi verrà utilizzato sempre questo secondo modello. Negli esperimenti fatti sono stati usati 256 *bins* di dimensione 1 (cioè l'istogramma massimo).

Nel caso generale di una bitmap a più canali (per esempio 3 canali a 8 bit) si tende a estrarre *un'istogramma indipendente per ogni canale*, secondo la stessa logica. Si ritiene talvolta opportuno cambiare la base dello spazio dei colori. Se per esempio viene effettuata la trasformazione $RGB \hookrightarrow HSB$ si ottiene un canale foriero dell'informazione *intensità* mentre le informazioni sul colore stanno negli altri due (esattamente come passare da coordinate cartesiane a sferiche: un solo campo riassume la grandezza del vettore e gli altri due la sua direzione). Si può così confrontare l'istogramma di un'immagine in bianco e nero con quello di un'immagine a colori. Nell'attuale esperienza sono state però usate solo immagini in bianco e nero, quindi il problema non si pone.

Si è accertato sperimentalmente che questa statistica è la più forte, stabile e rappresentativa di un'immagine, tanto che una possibile estensione consiste

nell'usare tutti gli N valori di $\mathcal{H}'$ come descrittori, in modo da poter creare delle metriche che sfruttino appieno l'istogramma (la letteratura è piena di funzioni di *matching* tra istogrammi).

Analogamente alle immagini, anche per l'istogramma si possono calcolare i momenti di ordine k (spaziali - M_k, centrali - μ_k, normali - η_k)[8]:

$$M_k[\mathcal{H}'] = \sum_{i=0}^{N-1} i^k \cdot \mathcal{H}'_k \tag{2.15}$$

$$\mu_k[\mathcal{H}'] = \sum_{i=0}^{N-1} (i - M_1)^k \cdot \mathcal{H}'_k \tag{2.16}$$

$$\eta_k[\mathcal{H}'] = \sqrt[k]{\frac{\mu_k}{M_0 \cdot N^k}} \tag{2.17}$$

Da notare che gli η_k sono stati normalizzati sia rispetto alla sommatoria (dividendo per M_0) sia rispetto ai valori che può assumere $g'(x,y)(\doteq \lfloor \frac{g(x,y)}{\xi} \rfloor))$, obbligando fondamentalmente g' ad avere valori tra 0 e 1. La radice rende addirittura *commensurabili* tra loro i momenti η_k.

Vediamo quali parametri sono stati estratti dall'istogramma:

$\mathcal{H}_\mu = M_1$ **(media)** É il più semplice e ovvio parametro; esprime il valor medio assunto dall'immagine.

$\mathcal{H}_\sigma = \sqrt{\mu_2}$, ***(scostamento quadratico medio)*** É un'importante misura di quanto il colore sia accentrato intorno al valor medio. Più è basso più il colore tende ad essere uniforme. Non è stato normalizzato in quanto così rappresenta il 'braccio d'inerzia' (nel solito isomorfismo meccanico) dell'istogramma, quindi – in buona misura – il numero di grigi 'coperti' intorno al valor medio. É considerato una buona stima dell'*omogeneità*[9] del colore (vedere però eq. (2.34) per una diversa definizione).

$\mathcal{H}_{Skew} = \eta_3$, ***(scostamento cubico medio normalizzato)*** In alcuni testi questo momento viene chiamato *Skewness*[10] (in realtà per alcuni la definizione

[8] con la differenza che le dimensioni passano da 2 a 1.

[9] attenzione però: alta varianza $\Longrightarrow$ bassa omogeneità.

[10] da *skew*, storto, sbilenco. In un testo italiano viene chiamata 'asimmetria', poiché vale zero

è quella di M_3, non di μ_3); il suo valore ci dice in che direzione è spostata l'eventuale 'gobba' del grafico rispetto al valor medio (vedi nota).

$\mathcal{H}_{Kurt}$, ***(scostamento quartico medio normalizzato)*** Non è altro che η_4. In alcuni testi questo momento viene chiamato *Curtosi (Kurtosis)* (ancora, essi talvolta usano M_4, non μ_4), in altri *flatness*; dovrebbe dare un'idea della piattezza del disegno. [11]

$\mathcal{H}_{MaxY} = \max_{k\in[0,N-1]} H'_k$, ***massima ordinata*** Dice qual è il valore massimo che l'istogramma assume (ovvero quanti pixel condividono uno stesso valore di grigio). Si è notato che questo valore è molto oscillante e poco significativo.

$\mathcal{H}_{MaxX} = H^{-1}(\mathcal{H}_{MaxY})$, **massima ascissa** Ci dice qual è il grigio più frequente nell'immagine. Questo valore è abbastanza stabile, e per quanto simile a $\mathcal{H}_\mu$ si mantiene su valori diversi ed è esso stesso discriminante.

2.5.2 Statistiche di grigio del II ordine

Rientrano in questa categoria tutte quelle statistiche volte a determinare la mutua *correlazione* tra i toni di grigio. Vediamo di chiarire con un esempio.

Una statistica di ordine zero potrebbe essere il valor medio di grigio di un'immagine. Per quanto sia un'ottima feature (che peraltro *è* usata), essa non distingue tra un corpo grigio ed un corpo per metà bianco e per metà nero.

Una statistica di ordine uno riesce invece ad accorgersi di questo fatto, distingue le due immagini appena dette ma fallisce a distinguere una finissima trama a pixel bianchi e neri alternati da un'altra immagine che contiene due soli blocchi, uno bianco l'altro nero.

Abbiamo dunque bisogno di una funzione in grado di mettere in relazione le

se la distribuzione è simmetrica intorno allo zero e positiva (negativa) se ha la 'gobba' a destra (sinistra).

[11] poiché infatti pondera con la quarta potenza peseranno maggiormente grandi deviazioni, a parità di μ e σ; fondamentalmente, rimanendo uguali i precedenti tre parametri, misura se nel grafico vi sono punte (valori alti) o altipiani (valori bassi). Per avere un ordine di grandezza, la distribuzione normale ha Curtosi 3.

informazioni sul colore in un modo più complesso, che metta in luce il modo in cui i colori *concorrono* nel costituire la tessitura del disegno.

La Matrice di co-occorrenza

Non ci si può esimere dal citare **Haralick** [8] quando si parla di *texture analysis*. Egli infatti ha creato un primo modello matematico molto semplice eppure molto potente con cui tutti si sono confrontati, riprendendolo, correggendolo, rendendolo più efficiente.[12] Questi ha definito una matrice di co-occorrenza[13] dei toni di grigio a partire dalla quale si possono estrarre una serie di *features* che rappresentino certi aspetti dell'immagine (contrasto, entropia, valor medio, ...)

Vediamo di spiegare con un esempio l'idea alla base della matrice.

Sia dato un operatore P_1 che recita:

$$P_1(A,B) \Longleftrightarrow \{\boldsymbol{B} \textit{ sta in basso a destra di un posto rispetto ad } \boldsymbol{A}\}, \tag{2.18}$$

dove A e B sono due punti dell'immagine. Sia dato inoltre P_2 secondo la seguente definizione:

$$\begin{aligned} P_1(A,B) \iff & \{\boldsymbol{B} \textit{ sta in basso a destra } o & (2.19)\\ & \textit{in alto a sinistra di un posto rispetto ad } \boldsymbol{A}\}, & (2.20)\end{aligned}$$

Sia data (a mo' d'esempio) un'immagine a 3 livelli di grigio $\mathcal{I}$ e un operatore P. Sarà univocamente definita la matrice 3x3 che contiene nella cella c_{ij} il numero (di occorrenze) di pixel del colore i^{mo} che siano in corrispondenza con un pixel del colore j^{mo} *secondo l'operatore* P.

Supponendo la matrice che rappresenta l'immagine essere (tanto per intenderci, potrebbe essere 0=bianco, 1=grigio, 2=nero):

[12] Dopo 30 anni, è considerato ancora il più potente strumento (o modello) di analisi della tessitura di un'immagine, e vari sono gli usi che se ne possono fare.

[13] Negli scritti di Haralick [8] viene definita *Matrice di Dipendenza Spaziale dei Livelli di Grigio (SGLDM)*; il nome *Matrice di co-Occorrenza dei Livelli di Grigio* va invece a Gonzales e Woods [6].

$$\mathcal{I} \equiv \begin{bmatrix} 0 & 0 & 0 & 1 & 2 \\ 1 & 1 & 0 & 1 & 1 \\ 2 & 2 & 1 & 0 & 0 \\ 1 & 1 & 0 & 2 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

e supponendo di usare gli operatori P_1 e P_2 sopra citati, le matrici A_1 e A_2 che vengono fuori saranno:

$$\mathbf{A_1} = \begin{bmatrix} 4 & 2 & 1 \\ 2 & 3 & 2 \\ 0 & 2 & 0 \end{bmatrix}, \qquad \mathbf{A_2} = \begin{bmatrix} 8 & 4 & 3 \\ 4 & 6 & 6 \\ 3 & 4 & 0 \end{bmatrix}$$

Da notare che se l'operatore è simmetrico anche la matrice risultante sarà simmetrica.

Segue una definizione un po' più rigorosa:

Definizione 2.5.1 (Matrice di co-occorrenza) *Sia data un'immagine $\mathcal{I}$ (definita su un dominio $\mathcal{D} \subseteq \mathbb{N}_W \times \mathbb{N}_H$) a valori in $\mathbb{N}^0_{G-1}$. Siano date inoltre una distanza positiva d e una direzione α. Definiamo* Matrice di Co-occorrenza*:*

$$\begin{aligned} P^{Har}_{d,\alpha}(g_1, g_2)[\mathcal{I}] &\equiv [p_{g_1,g_2}], \qquad \textit{dove:} \\ p_{g_1,g_2} &\doteq Card\Big\{ (\mathbf{P_1}, \mathbf{P_2}) \in \mathcal{D}^2 \qquad t.c. \\ &\qquad g(\mathbf{P_1}) = g_1,\ g(\mathbf{P_2}) = g_2,\ \mathbf{P_2} - \mathbf{P_1} = d \cdot \overrightarrow{\alpha} \Big\}, \quad (2.21) \end{aligned}$$

dove $\mathcal{D}$ è il dominio di definizione dell'immagine (la maschera), $g(\mathbf{P})$ è la funzione che associa al pixel $\mathbf{P}$ il suo gray-level, $\overrightarrow{\alpha}$ il versore che ha per direzione α. Così costruita la matrice è il più generale possibile e può *essere asimmetrica.*

Quindi, la matrice associa a ogni cella nella posizione (g_1, g_2) il numero di pixel aventi tonalità di grigio g_1 che vedono nella direzione α a distanza d un pixel *appartenente al dominio di definizione* con tonalità g_2.

In casi pratici si usano soltanto le 4 direzioni 0°, 45°, 90° e 135° (ricordiamo che ogni pixel ha 8 vicini, e ogni altra direzione andrebbe calcolata fornendo dunque un dato non effettivo ma *interpolato*).

Per quel che riguarda la **simmetria** degli operatori, c'è una piccola inconsistenza in letteratura [19]; secondo alcuni testi [16] la SGLDM (*Spatial Gray-Level Dependency Matrix*) andrebbe calcolata per un *generico* operatore e quindi *può* essere asimmetrica; per altri [8] la matrice è intrinsecamente simmetrica poiché per qualunque operatore P si prende l'operatore P' definito come:

$$P'(A,B) \longleftarrow P(A,B) \vee P(B,A) \tag{2.22}$$

Definizione 2.5.1 (Operatore simmetrico) *Un operatore P_a si dice* simmetrico se esiste un operatore P_b di cui P_a è la versione simmetrica.

(Da notare che nell'esempio P_2 è proprio la 'chiusura simmetrica' di P_1.) Nella definizione di Haralick l'operatore che associa a ogni punto P_1 il punto P_2 è **simmetrico** e quindi ogni coppia (distanza, direzione) fornisce due possibili scostamenti nei due versi possibili (quindi $45° \equiv 225°$, e così via). Si è tentato di dare una definizione il più generale possibile, sebbene nel corso del testo vengano sempre usate matrici simmetriche (mentre i parametri di Unser non nasceranno necessariamente da operatori simmetrici, vedi pag. 32). Due interessanti proprietà sono:

- Sia P_2 la chiusura simmetrica di P_1 e P_1 un operatore semplice (che associa a ogni punto uno ed un solo punto); siano M_2 e M_1 (rispettivamente) le loro matrici di co-occorrenza. Allora vale:

$$M_2 = M_1 + M_1^T \tag{2.23}$$

- Come si vede dalla eq. (2.23), se un operatore è simmetrico, la sua matrice di co-occorrenza è simmetrica. Si noti che questo diminuisce l'insieme di *features* significative per la matrice (come si può vedere più avanti (pag. 28), alcune features diventano uguali a due a due, e certi vettori diventano simmetrici). Comunque, nessuna delle 14 features di Haralick risente della simmetria della matrice (nel senso che nessuno dei 14 valori perde di significato).

Le matrici più usate sono le quattro matrici simmetriche che si ottengono ponendo la distanza a 1 (certe *textures* hanno caratteristiche per cui è necessario

spingersi oltre con lo studio, magari usando più matrici per distanze crescenti). Sommandone i valori si ottiene una nuova matrice mediata su tutte le direzioni (che quindi perde molte informazioni sulla direzione ma guadagna in isotropia):

$$P_{\Sigma}^{Har} = \frac{P_{1,0^\circ}^{Har} + P_{1,45^\circ}^{Har} + P_{1,90^\circ}^{Har} + P_{1,135^\circ}^{Har}}{4}, \tag{2.24}$$

dalla cui normalizzazione si ottiene la matrice solitamente più usata per estrarre le *features di Haralick*.

$$P^{Har} = \frac{P_{\Sigma}^{Har}}{\sum_{g_1=0}^{G-1}\sum_{g_2=0}^{G-1} P_{\Sigma}^{Har}(g_1, g_2)}, \tag{2.25}$$

dove il denominatore può essere visto (oltre che, semplicemente, come la somma di tutti i valori delle celle della matrice) come l'insieme di tutte le possibili coppie di pixel secondo la relazione indotta da (d, α). Questo numero *non* è semplicemente l'area dell'immagine (come si potrebbe pensare), poiché nei bordi certe coppie sono proibite.

Ancora una volta, per ridurre l'esosità dei calcoli e l'occupazione di memoria, si tende a ridurre la matrice ingrossando i *bin* e diminuendone drasticamente la dimensione.

Per poter calcolare il vettore di features proposto da Haralick occorre prima fare delle pre-elaborazioni sulla matrice (più per comodità di notazione che per concetto). La matrice su cui si sono svolti i calcoli usa solo 16 bin (riducendo la matrice da 64K a 256 celle) ed è simmetrica.

$$
\begin{aligned}
p_x(g) &= \sum_{g'=0}^{G-1} p(g,g') \qquad p_y(g') = \sum_{g=0}^{G-1} p(g,g') \\
p_{x+y}(g'') &= \sum_{g=0}^{G-1} \left(\sum_{\substack{g'=0 \\ |g+g'|=g''}}^{G-1} p(g,g') \right), \qquad g'' \in 0..2G-2 \qquad (2.26) \\
p_{x-y}(g'') &= \sum_{g=0}^{G-1} \left(\sum_{\substack{g'=0 \\ |g-g'|=g''}}^{G-1} p(g,g') \right), \qquad g'' \in 0..G-1 \qquad (2.27) \\
\mu_x(g) &= \sum_{g=0}^{G-1} p_x(g) \qquad \mu_y(g') = \sum_{g'=0}^{G-1} p_y(g') \\
\sigma_x(g) &= \sqrt{\sum_{g=0}^{G-1} p_x(g)(g-\mu_x)^2} \qquad \sigma_y(g') = \sqrt{\sum_{g'=0}^{G-1} p_y(g')(g-\mu_y)^2} \\
HXY &= -\sum_{g=0}^{G-1}\sum_{g'=0}^{G-1} p(g,g') \operatorname{logz}(p(g,g')) \qquad (2.28) \\
HXY1 &= -\sum_{g=0}^{G-1}\sum_{g'=0}^{G-1} p(g,g') \operatorname{logz}(p_x(g)p_y(g')) \\
HXY2 &= -\sum_{g=0}^{G-1}\sum_{g'=0}^{G-1} p_x(g)\, p_y(g') \operatorname{logz}(p_x(g)p_y(g')) \\
HX &= -\sum_{g=0}^{G-1} p_x(g) \operatorname{logz}(p_x(g)) \\
HY &= -\sum_{g'=0}^{G-1} p_y(g') \operatorname{logz}(p_y(g')) \\
\operatorname{logz} &:= \begin{cases} 0, & x = 0; \\ log_e(x), & x > 0; \end{cases} \qquad (2.29)
\end{aligned}
$$

A partire da queste definizioni possiamo calcolare le *features di Haralick* cominciando dalle più importanti:

Momento Angolare Secondo . É la somma dei quadrati dei valori della matrice. É una feature molto utile che viene usata in molti articoli consultati. É anche nota come *Uniformità*. Questo valore è piccolo quando tutti gli

elementi della matrice sono simili, quindi più alto è il valore, più irregolare è la matrice. Unser lo chiama **energia** [10].[14]

$$\mathbf{F}_{ASM}^{Har} = \sum_{g=0}^{G-1}\sum_{g'=0}^{G-1} p^2(g,g') \tag{2.30}$$

Contrasto . Esprime il contrasto. Può essere interpretata come una media pesata su tutte le diagonali parallele alla principale che premi di più quelle lontane dalla diagonale (premiando quindi la correlazione - nel nostro caso la vicinanza - tra tonalità diverse, ovvero in contrasto tra loro). Viene anche detto *Inerzia* o *Momento Differenza*, proprio perché premia di molto gli elementi distanti dalla diagonale principale (quindi pixel vicini con forte contrasto di colore). Ammette come range $[0,(n-1)^2]$, dove n è la dimensione della matrice.

$$\mathbf{F}_{Con}^{Har} = \sum_{g=0}^{G-1} g^2 \left(\sum_{\substack{g'=0 \\ |g'-g''|=g}}^{G-1}\sum_{g''=0}^{G-1} p(g',g'') \right) \tag{2.31}$$

Correlazione . É fondamentalmente un Momento Centrale Normalizzato η_{11}, visto in precedenza. Differisce da esso solo per quanto riguarda il fattore di normalizzazione $(\sigma_x\sigma_y)$ e la 'centralizzazione' $(\mu_x\mu_y)$.

$$\mathbf{F}_{Cor}^{Har} = \frac{1}{\sigma_x\sigma_y}\left(\sum_{g=0}^{G-1}\sum_{g'=0}^{G-1} (gg')p(g,g') - \mu_x\mu_y \right) \tag{2.32}$$

Somma dei quadrati: Varianza Purtroppo non è stata riscontrata una definizione coerente né in [8] né in [4] quindi questa feature non è semplicemente stata usata.[15]

$$\mathbf{F}_{SSV}^{Har} = \sum_{g=0}^{G-1}\sum_{g'=0}^{G-1} (g-\mu)^2 p(g,g') \tag{2.33}$$

[14]effettivamente nello studio dei segnali si usa chiamare *potenza istantanea* di un segnale $u(t)$ il suo quadrato $P(t) = u^2(t)$ ed *energia* la somma nel tempo: $E_{tot} = \int u^2(\tau)d\tau$.

[15]Ha infatti senso solo per una matrice simmetrica. Sebbene siano state usate effettivamente solo matrici simmetriche , non si è trovato giusto usare formule che lo richiedessero a priori, riducendone quindi il campo d'applicazione.

Momento Differenza Inverso Ricorda vagamente il contrasto, con la differenza che pesa *meno* le diagonali che si allontanano dalla principale. La filosofia è quindi *contraria* al contrasto (detto anche *Momento Differenza...*).É detto anche **Omogeneità** ([10], [19]). Misura dunque quanto pixel vicini abbiano toni di grigio omogenei tra loro.

$$\mathbf{F}_{IDM}^{Har} = \sum_{g=0}^{G-1}\sum_{g'=0}^{G-1} \frac{p(g,g')}{1+(g-g')^2} \tag{2.34}$$

Media della Somma É la media del vettore p_{x+y} (eq. (2.26)), che a sua volta contiene le somme di tutte le diagonali ortogonali alla principale.

$$\mathbf{F}_{SAv}^{Har} = \sum_{g=0}^{2G-2} g\, p_{x+y}(g) \tag{2.35}$$

Varianza della Somma Analogamente alla precedente, è una stima del secondo ordine del vettore p_{x+y} centralizzata rispetto alla media eq. (2.35).

$$\mathbf{F}_{SVa}^{Har} = \sum_{g=0}^{2G-2} g\, p_{x+y}(g) \tag{2.36}$$

Entropia della Somma Ancora una volta fornisce una misura del vettore p_{x+y}, questa volta l'*entropia* (ovvero una misura di disordine del vettore stesso).

$$\mathbf{F}_{SEn}^{Har} = \sum_{g=0}^{2G-2} p_{x+y}(g)\,\mathrm{logz}(p_{x+y}(g)) \tag{2.37}$$

Entropia Questa, a differenza della eq. (2.37), è una misura di entropia della matrice nella sua globalità. Ammette come range $[0, 2\,log(n)]$, dove n è la dimensione della matrice. Vedi eq. (2.28).

$$\mathbf{F}_{Ent}^{Har} = HXY = -\sum_{g=0}^{G-1}\sum_{g'=0}^{G-1} p(g,g')\,\mathrm{logz}(p(g,g')) \tag{2.38}$$

Varianza della Differenza Esprime la varianza del vettore p_{x-y}, che a sua volta (eq. (2.27)) è un vettore che ha per elementi la somma dei valori della matrice visti per diagonali *parallele* alla diagonale principale.

$$\mathbf{F}_{DVa}^{Har} = \sum_{g=0}^{2G-2} p_{x-y}(g)\,\mathrm{logz}(p_{x-y}(g)) \tag{2.39}$$

Entropia della Differenza Misura l'entropia del vettore p_{x-y}, (eq. (2.27), pag. 28).

$$\mathbf{F}_{DEn}^{Har} = \sum_{g=0}^{G-1} p_{x-y}(g) \operatorname{logz}(p_{x-y}(g)) \tag{2.40}$$

Misure d'Informazione sulla Correlazione Seguono due descrittori generali che nascono da 5 misure di *Entropia* (eq. (2.28) e oltre) della Matrice di Cooccorrenza per cui si rimanda a pag. 28.

$$\mathbf{F}_{IC1}^{Har} = \frac{HXY - HXY1}{\max\{HX, HY\}} \tag{2.41}$$

$$\mathbf{F}_{IC2}^{Har} = \sqrt{1 - e^{-2|HXY2 - HXY|}} \tag{2.42}$$

Massimo Coefficiente di Correlazione Per completezza si segnala l'esistenza di questa 14^{ma} feature che è stata evitata per un compromesso costo-resa. Vediamo comunque la sua definizione:

$$\mathbf{F}_{MCC}^{Har} = \sqrt{secondo\, più\, grande\, autovalore\, di\, Q}, \tag{2.43}$$

$$dove\ Q = [q(g, g')] := \sum_{g''=0}^{G-1} \frac{p(g, g'')p(g', g'')}{p_x(g)p_y(g')}$$

Dall'analisi si altri testi vengono di seguito riportate altre semplici features; sono state usate solo in piccola parte ma vengono riportate tutte per completezza di trattazione:

Pseudo-Omogeneità . Assomiglia molto a $\mathbf{F}_{IDM}^{Har}$, eq. (2.34), con la differenza che il peso non è di tipo $\frac{1}{1+\delta^2}$ bensì $\frac{1}{1+\delta}$. In certi testi viene chiamato omogeneità. La maggior parte dei riferimenti trovati danno però quel nome all'eq. (2.34), quindi è stato adottato un nuovo nome per *questo* descrittore. La filosofia, d'altronde, è la stessa, cambia solo il peso.

$$\mathbf{F}_{OMO}^{Oth} = \sum_{g=0}^{G-1} \sum_{g'=0}^{G-1} \frac{p(g, g')}{1 + |g - g'|} \tag{2.44}$$

Cluster Shade . É un momento centrale di ordine 3 un po' particolare; è stato

proposto da Unser [10].

$$\mathbf{F}_{CLS}^{Oth} = \sum_{g=0}^{G-1}\sum_{g'=0}^{G-1}(g+g'-2\mu)^3 p(g,g')^{16} \tag{2.45}$$

Cluster Prominence . É un momento centrale di ordine 4; anch'esso è stato proposto da Unser [10].

$$\mathbf{F}_{CLP}^{Oth} = \sum_{g=0}^{G-1}\sum_{g'=0}^{G-1}(g+g'-2\mu)^4 p(g,g') \tag{2.46}$$

La matrice di co-occorrenza si presta bene, tra l'altro, anche come *edge detector*.

Unser: gl'istogrammi somma e differenza

Tra tanti approcci alternativi ad Haralick è stato scelto in questo *iter* Unser [10] per la semplicità del modello (non altrettanto lo sono le sue giustificazioni statistiche!), l'efficienza, la somiglianza con Haralick e soprattutto la capacità discriminante.

Unser ha cercato di rendere il metodo di Haralick meno gravoso sia in termini di tempo di computazione che di spazio occupato. Da una ricerca statistica, Unser ha tratto che le due variabili che compaiono nella matrice di co-occorrenza $P(g,g')$ hanno la stessa media e varianza e sono scorrelate.[17] Poiché *"La somma e la differenza di due variabili casuali con le stesse varianze sono scorrelate e definiscono gli assi principali della loro funzione associata di probabilità congiunta.[...]"*, egli non ha fatto altro che 'percorrere la matrice di Haralick secondo la nuova base:

$\mathcal{B}_{Uns} \equiv ((\frac{1}{\sqrt{2}},\frac{1}{\sqrt{2}})^T,(\frac{1}{\sqrt{2}},\frac{-1}{\sqrt{2}})^T)$

[16] Dove si è trovato μ si è supposto che l'Autore desse per scontata la simmetria della matrice (quindi $\mu := \mu_x = \mu_y$). Non sono stati usati descrittori che *supponessero* la simmetria della matrice.

[17] il problema *può* apparire mal posto, in quanto non si è mai parlato di probabilità; effettivamente la matrice di Haralick ben si presta ad un'interpretazione statistica come distribuzione di probabilità a due variabili, se la si vede come la *probabilità che un colore g_1 sia vicino a g_2*, ove con *vicino* s'intende rispetto a un certo operatore P.

Definite dunque le due nuove variabili:

$$\begin{cases} z_1 &= \frac{1}{\sqrt{2}}(y_1 + y_2) \\ z_2 &= \frac{1}{\sqrt{2}}(y_1 - y_2) \end{cases}$$

queste si trovano scorrelate tra loro (la loro covarianza è nulla). Quando due variabili aleatorie sono scorrelate e indipendenti, la funzione di probabilità congiunta può essere calcolata come segue:

$$P(y_1, y_2) = P(z_1, z_2) \cong P_s(z_1) \cdot P_d(z_2) \tag{2.47}$$

Purtroppo la seconda uguaglianza (sebbene sempre valida per variabili aleatorie gaussiane) viene soddisfatta solo per variabili aleatorie *indipendenti*; la scorrelazione è una condizione necessaria ma non sufficiente per l'indipendenza. Unser si spinge però oltre con considerazioni sul fatto che quell'uguaglianza (eq. (2.47)), sebbene non vera, fornisca un'approssimazione la cui bontà può essere calcolata.

$$P_y(g, g') = k \cdot P_s(g + g') \cdot P_d(g - g') \cong P(g, g'), \tag{2.48}$$

dove k è un'opportuna costante di normalizzazione. Ricapitolando, Unser propone di approssimare le G^2 celle della matrice di Haralick usando due soli vettori (gli istogrammi somma e differenza, appunto) e calcolandone i valori all'occorrenza. L'errore commesso nell'approssimare non tanto i valori quanto la distribuzione è stimabile con l'entropia relativa:

$$I(P, \hat{\mathrm{P}}) = H_s + H_d - H_y - log(k) \geq 0 \tag{2.49}$$

Questo dato è molto interessante poiché si può vedere *praticamente* se 'Unser riesce ad approssimare bene Haralick' per una certa immagine. Si può notare che il costo di questa prova è pressoché nullo poiché nella eq. (2.49) i primi due valori sono due importanti Features degl'istogrammi somma e differenza, il terzo non è altro che la eq. (2.28), pag. 28, il quarto dato da: $\frac{1}{\sum_{g=0}^{G-1}\sum_{g'=0}^{G-1}\hat{\mathrm{P}}_{(g,g')}}$.

La definizione formale dei cosiddetti "istogramma somma" e "istogramma differenza" è la seguente:

$$\begin{aligned} \mathbf{h}_s(g) &= Card\{(\mathbf{P_1}, \mathbf{P_2}) \in \mathcal{D}^2 \,|\, g(P_1) = g_1, g(P_2) = g_2, \\ &\quad g = g_1 + g_2, \mathbf{P_2} - \mathbf{P_1} = d \cdot \overrightarrow{\alpha}\} \end{aligned} \tag{2.50}$$

$$\begin{aligned} \mathbf{h}_d(g) &= Card\{(\mathbf{P_1}, \mathbf{P_2}) \in \mathcal{D}^2 \,|\, g(P_1) = g_1, g(P_2) = g_2, \\ &\quad g = g_1 - g_2, \mathbf{P_2} - \mathbf{P_1} = d \cdot \overrightarrow{\alpha}\} \end{aligned} \tag{2.51}$$

Anche qui è poi utile una normalizzazione. Sia N_{tot} la somma dei valori di uno dei due istogrammi (è infatti uguale per entrambi). La versione normalizzata degli istogrammi sarà:

$$\hat{\mathrm{P}}_s(i) = \frac{\mathbf{h}_s(i)}{N_{tot}}; \qquad \hat{\mathrm{P}}_d(j) = \frac{\mathbf{h}_d(j)}{N_{tot}} \tag{2.52}$$

Ancora una volta, dunque, un istogramma è definito da una distanza d e da una direzione α; Unser propone di calcolare tutti gli 8 istogrammi possibili a distanza 1 (4 direzioni per 2 tipi d'istogramma). Per ciascuno di essi propone 4 feature (portando così il numero di descrittori a 32). In questo lavoro sono stati usati gl'istogrammi e le features consigliate da Unser applicati a distanza 1; per quanto riguarda la direzione, invece, dei suoi 4 sono stati considerati solo quello orizzontale e verticale, ed è stato infine aggiunto quello mediato sulle 4 direzioni per poter avere materia di confronto con i descrittori di Haralick (calcolati sulla matrice pag. 27).

Seguono le quattro feature di Unser:

1. **Media** É semplicemente il valor medio assunto dall'istogramma. Si noti che nel caso della versione simmetrica di Unser (che è stata ottenuta sommando i 4 istogrammi di Unser nelle 4 direzioni e normalizzando) l'istogramma differenza ha sempre media nulla.[18] Questo è necessario, ma gli altri valori dell'istogramma differenza sembrano rimanere significativi.

$$\mathbf{F}^{Uns}_{MED} = \sum_{g \in \mathcal{D}} g\, \hat{\mathrm{P}}(g) \tag{2.53}$$

2. **Energia** Altri non è che il *Momento Angolare Secondo*, già definito in eq. (2.30), pag. 29.

$$\mathbf{F}^{Uns}_{ASM} = \sum_{g \in \mathcal{D}} \hat{\mathrm{P}}^2(g) \tag{2.54}$$

In particolare Unser mostra che, nell'ipotesi che valga la eq. (2.47), vale la seguente:

$$\mathbf{F}^{Uns}_{ASM}[H_s] \cdot \mathbf{F}^{Uns}_{ASM}[H_d] = \mathbf{F}^{Har}_{ASM} \tag{2.55}$$

[18] poiché è simmetrica la matrice di co-occorrenza e questa somma la percorre secondo la diagonale secondaria.

3. **Contrasto** Definizione del tutto simile all'eq. (2.31), pag. 29. Esiste una relazione con il contrasto di Haralick, ma non è banale.

$$\mathbf{F}_{CON}^{Uns} = \sum_{g \in \mathcal{D}} (g - \mu)^2 \hat{\mathrm{P}}(g) \tag{2.56}$$

4. **Entropia** Definizione del tutto simile a eq. (2.31), pag. 29.

$$\mathbf{F}_{ENT}^{Uns} = -\sum_{g \in \mathcal{D}} \hat{\mathrm{P}}(g) \operatorname{logz}(\hat{\mathrm{P}}(g)) \qquad [19] \tag{2.57}$$

In particolare Unser mostra che, nell'ipotesi che valga la eq. (2.47), vale la seguente semplice relazione:

$$\mathbf{F}_{ENT}^{Uns}[H_s] + \mathbf{F}_{ENT}^{Uns}[H_d] = \mathbf{F}_{ENT}^{Har} \tag{2.58}$$

Da notare che sono stati fatti oscillare gli indici su D_h (dominio dell'istogramma) per comodità di notazione: i due istogrammi hanno infatti la stessa dimensione ma *offset* diverso: h_s varia su $[0, 2G-2]$, mentre h_d varia su $[-G+1, G-1]$.

Con questo approccio, Unser mantiene un istogramma per ogni direzione, a differenza di Haralick che usa la matrice mediata sulle varie direzioni. Evidentemente un algoritmo più leggero – sia perché gl'istogrammi occupano meno, sia perché vengono estratte meno features (4+4 contro 14) – consente di poter memorizzare più istogrammi, a parità di carico computazionale consentito.

2.6 Conclusioni

Si è mostrata finora una grande varietà di descrittori di forma, colore e trama. In questa sezione si cercherà di giustificare alcune scelte.

- Sono state scartate a priori *features* che potevano essere calcolate solo su immagini rettangolari, come per esempio la trasformata di Fourier [5]. Infatti le immagini usate sono spesso così piccole e la loro maschera è a sua volta così piccola che considerare lo sfondo parte dell'oggetto altererebbe troppo la misura. Fortunatamente le matrici di co-occorrenza e i momenti delle scale di grigio non hanno questo problema.

[19] Per la definizione di *logz*, vedi eq. (2.29), pag. 28.

- Può apparire paradossale che siano state impiegate *sia* le features di Haralick *sia* quelle di Unser, dato che le seconde sono nate per approssimare le prime. Ebbene, sebbene alcune di esse siano risultate sperimentalmente identiche (il che è stato un sollievo sia per la corretta implementazione che per le assunzioni di Unser), c'è da dire un paio di cose:

  1. Entrambi i modelli (nell'attuale implementazione) hanno informazioni che l'altro non dà.
  2. Nelle prove fatte si è scoperto con sorpresa che (in alcuni casi) certi parametri di Unser discriminavano con successo due immagini quando il loro corrispondente parametro di Haralick non se ne accorgeva.
  3. L'utilizzo della matrice di co-occorrenza viene consigliato su un'*unica* matrice, in particolare non ha senso usare altri che la matrice definita in eq. (2.24), pag. 27, magari con distanze superiori a 1. Al contrario per Unser gl'istogrammi mediati sulle 4 direzioni (che sono stati effettivamente usati) sono in generale meno significativi e discriminanti.

  Diciamo quindi che le feature di Haralick sono isotrope e invarianti per rotazione; quelle di Unser, invece, sono sensibili alla direzione e costituiscono un insieme, se non indipendente, almeno più *coprente* delle singole parti.

- In un lavoro off-line ci si può permettere di fare a volte calcoli ridondanti, in favore di una migliore comprensione del problema; s'intende in una futura integrazione on-line del sistema di tagliare completamente i calcoli di Haralick, che vengono stimati essere *molto* [20] più costosi del feature-set di Unser.

- I descrittori più stabili sono senz'altro quelli derivanti dall'istogramma di toni di grigio (che vanta il maggior numero di invarianze *effettive*), l'elongazione, i descrittori locali di grigio (valore medio del pixel centrale, ...). Sono invece risultati meno stabili di quanto si potesse prevedere i momenti e alcune feature di tessitura.

[20] in [4] il rapporto tra i tempi è 10.5 volte.

Capitolo 3

Calcolo di descrittori su una sequenza di immagini

> One of the most interesting aspects of the world is that it can be considered to be made out of patterns. A pattern is essentially an arrangement. It is characterized by the order of the elements of which it is made, rather than by the intrinsic nature of these elements.
>
> *Norbert Wiener*

3.1 Modelli per sequenze di immagini

Riprendendo la definizione Def. (2.1.1), pag. 11, possiamo definire una sequenza nel seguente modo:

Definizione 3.1.1 (Sequenza) *Una sequenza $\mathcal{S}_\tau$ è una successione di immagini $\mathcal{I}_{k,[X,Y,G]}$ lunga $\tau + 1$ (adotteremo infatti la numerazione $[0, \tau]$), con il vincolo implicito che, come già detto (pag. 11), G sia uguale per ogni istanza $\mathcal{I}_k$.*

Non è garantito, invece, che il supporto $\mathcal{D}$ delle immagini sia uguale per immagini successive, essendo una proprietà della *singola* immagine.

Si parlerà spesso di *History* dando una connotazione temporale a una *sequenza*, senza però alterarne il modello. Si assume dunque che la sequenza *nasca* nell'immagine 0, e che volta per volta la più recente sia l'N^{ma}; questo nasce dal

fatto che, in pratica, si acquisiscono le immagini *real-time* e la sequenza cresce a poco a poco mantenendo fisse le immagini precedenti e aggiungendone una per ogni *frame* elaborato. In particolare, quando si parlerà di *history*, si alluderà non tanto alla sequenza stessa ma alla struttura numerica associata ad essa (derivante dal calcolo di descrittori). Per una migliore trattazione, vedere (Def. (3.3.2), pag. 44).

3.2 Calcolo di Descrittori per un'Immagine

Cominciamo con una necessaria definizione.

Definizione 3.2.1 (Descrittore) *Un descrittore (o* feature*)* φ *è una generica funzione* $\varphi : \mathcal{S}_{\mathcal{I}_G} \longmapsto \mathrm{R}$.[1]

Un descrittore, insomma, accetta in ingresso un'immagine ed estrapola da essa un parametro che, appunto, la descriva secondo la semantica della *feature* stessa (vedi Cap. 2 per una completa trattazione dei descrittori usati). Inutile dire che la bontà di un descrittore può essere misurata facilmente su una *sequenza*: per comportarsi bene dovrà essere molto stabile per una sequenza e possibilmente essere discriminante per sequenze diverse.

Definizione 3.2.2 (Pattern) *Data un'immagine* $\mathcal{I}(\in \mathcal{S}_{\mathcal{I}_G})$ *e una lista* ordinata *di* λ *descrittori* $\boldsymbol{\varphi} := (\varphi_1, ..., \varphi_\lambda)$*, definiamo il* **vettore di features** $\boldsymbol{v}[\mathcal{I}]$ *(o* **pattern***) secondo la semplice relazione:*

$$\boldsymbol{v}[\mathcal{I}] := (\varphi_1(\mathcal{I}), ..., \varphi_\lambda(\mathcal{I})) \tag{3.1}$$

Il *pattern* è, molto semplicemente, una lista ordinata di numeri reali corrispondente alla 'firma' dell'immagine vista dai λ descrittori.

Nel lavoro svolto sono stati usati 47 descrittori per ogni immagine. Ci si potrebbe chiedere perché ne siano stati usati così tanti. Effettivamente all'inizio il lavoro è consistito nel provare *tanti* descrittori e vedere quali meglio si prestavano a descrivere le immagini per poter arrivare dove il modulo di Tracking non

[1] Nell'attuale implementazione, i descrittori hanno sempre valori *non-negativi*. Questo per motivi statistici.

poteva arrivare neanche volendo, vista la sua visibilità del problema. Il tempo ha però rivelato (attraverso certi esperimenti) che certi descrittori apparentemente fallimentari riuscivano a discriminare tra due immagini molto simili quando altri non vi riuscivano. Si ricordi per giunta che, all'interno di ogni famiglia, il costo di estrazione di una sola feature o di tutte le features della famiglia stessa è molto simile. A poco a poco è andata maturando una nuova idea: usare quanti più descrittori possibile e sfruttare volta per volta quelli che meglio discriminavano tra due immagini.

Da notare che per una sola istanza d'immagine, dati λ descrittori, *non si può far altro* che calcolare λ valori; non si possono estrapolare statistiche da essi, poiché *in generale* non ha senso confrontare le feature *tra loro*, e a questo livello è precluso sapere quali relazioni ci possano essere tra due feature (potrebbero essere due entropie, ma anche un'entropia e un grigio medio...).

3.3 Calcolo di Descrittori per una *Sequenza*

Data una sequenza d'immagini $\mathcal{S}_\tau$ (Def. (3.1.1)) e una lista di descrittori $\boldsymbol{\varphi}$, l'insieme di possibilità che ci si aprono davanti è notevole: possiamo sicuramente calcolare, per ogni immagine, i λ descrittori, ma possiamo anche andare oltre e costruire una statistica per quella sequenza. In generale, si possono seguire due vie diverse:

1. Si può tenere in mente, per ogni immagine $\mathcal{I}_k$, il vettore $\boldsymbol{v}[\mathcal{I}]$; **non c'è *alcuna* perdita d'informazione**; se la sequenza cresce, però, quest'informazione occuperà una memoria proporzionale a λ e al tempo di vita della sequenza.[2]

2. Un altro approccio consiste nel tenere una statistica (di grandezza costante) che viene aggiornata all'arrivo di ogni nuovo frame. Poiché la dimensione della statistica non cresce nel tempo, occorre decidere oculatamente i parametri di cui tenere conto. Si ricorda che, volendo tenere a mente l'insieme *infinito* di momenti per ogni descrittore, potremmo mantenere comunque

[2]Nell'attuale implementazione, l'occupazione per ogni frame è $8 * 47$ bytes, quindi per una sequenza di 10 secondi a 25 frame/sec è di circa $80KB$ *per ogni* oggetto da tracciare.

tutta l'informazione. Sebbene ciò sia impensabile, è comunque molto interessante notare come i momenti dei primi ordini siano così significativi, rispetto alla limitata occupazione di memoria.[3]

L'approccio scelto è il secondo; anzitutto ha il notevole vantaggio di suddividere il carico computazionale nel tempo: ad ogni nuova immagine che entra a far parte della sequenza, oltre a calcolare i λ descrittori, si perde un po' di tempo ad aggiornare le statistiche; quando poi diventa necessario usarle per confrontare questa sequenza con altri oggetti (immagini o sequenze), la maggior parte dei calcoli è già stata fatta e non occorre ricalcolare nulla. Non che il *matching* abbia costo nullo, ma è ben altra cosa che confrontare *ex novo* due intere storie di *vettori di features*. Quando si esegue il matching, dunque, non bisogna calcolare alcun descrittore nuovo perché tutta l'informazione è già nella History. Per definirla, dobbiamo prima dare una definizione di *statistica*, intesa come l'insieme di informazioni che caratterizzano *un singolo descrittore*.

3.3.1 Statistiche associate a *un singolo* descrittore

Per ogni descrittore, si è deciso di estrarre un insieme di parametri (grazie alla disponibilità di un intero campione di valori) significativi a descriverlo. Una spiegazione esaustiva per ciascuno di essi segue (per comodità d'esposizione) la definizione.

Definizione 3.3.1 (Statistica) *Data una sequenza $\boldsymbol{\mathcal{S}_\tau}$ e un descrittore φ_n, definiamo la sua statistica $\boldsymbol{\varsigma_n}$ come un* **vettore riga** *di 11 caratteristiche:*

$$\varsigma(\mathcal{S}_\tau, \varphi_n) =: \boldsymbol{\varsigma_n}(\mathcal{S}_\tau) \equiv (\tau, \mu_{exp}, \sigma_{exp}, M_1, M_2, M_3, M_4, \delta, x_{max}, x_{min}, \ell) \qquad (3.2)$$

Non è in generale possibile calcolare la *statistica* di una qualunque sequenza ex novo[4], ma la si può calcolare in maniera *incrementale*. Cioé esiste una funzione

[3] I momenti possono essere visti come la parte più pregiata dell'informazione: sono un elemento di memoria formidabile e sono ancor più preziosi a fini statistici, ammesso che vengano eseguite le opportune normalizzazioni. Purtroppo da un punto di vista implementativo questi tendono a mandare in fretta in overflow la memoria dedicata ad essi; occorre dunque prestare attenzione alla precisione di somme con numeri elevati alla n^{ma} potenza...

[4] solo da un punto di vista implementativo: poiché a ogni nuova immagine che arriva perdo i dati sulle precedenti, posso calcolare la *statistica* solo per sequenze lunghe uno.

che, data la storia fino ad ora, $\mathcal{S}(1..\tau)$, e l'immagine successiva, $\mathcal{I}(\tau+1)$, restituisce la storia *aggiornata* al tempo $(\tau+1)$. Insieme alle definizioni che seguono, dunque, non verrà data la definizione *assoluta* di una generica caratteristica γ, quanto piuttosto la **definizione *induttiva***, secondo il seguente stile:

$$\gamma[\mathcal{S}_0] = f_0(\varphi_k(\mathcal{I}_0)) \tag{3.3}$$

$$\gamma[\mathcal{S}_{\tau+1}] = f_{ind}(\mathcal{S}_\tau,\ \varphi_k(\mathcal{I}_{\tau+1})), \tag{3.4}$$

dove la eq. (3.3) esprime il calcolo del descrittore al primo passo (in cui la sequenza è una semplice immagine), mentre la eq. (3.4) esprime il calcolo avendo a disposizione una sequenza e un'immagine singola.

Si può notare che, per come è stata definita la *statistica*, è *sempre* possibile calcolare quella nuova a partire dalla precedente accompagnata da una nuova immagine (se invece avessimo usato come caratteristiche μ e σ, per esempio, non avremmo potuto aggiornarle senza sapere il tempo di vita τ...)

Note che siano la sequenza $\mathcal{S}_\tau$ e il descrittore φ_n, verrà adottata per *brevità* la seguente notazione:

- si chiamerà l'insieme di valori che esso assume nel tempo col nome $(x_0, ..., x_\tau)$.
- si chiamerà $\mathcal{S}_{\tau+1}$ col nome S_{new};
- si chiamerà $\mathcal{S}_\tau$ col nome S_{old};
- si chiamerà il più recente x_n col nome x_{new}.
- si metterà il pedice $_{new}$ e $_{old}$ a fianco di una certa caratteristica k, intendendo rispettivamente il valore in $\mathcal{S}_{\tau+1}$ (che tipicamente andrà calcolato) e $\mathcal{S}_\tau$ (cui tipicamente ci si riferirà per fare il calcolo).

1. **Tempo di vita, τ** É la durata della sequenza (intesa come numero di immagini che essa contiene). Attenzione che una sequenza $\mathcal{S}_\tau$ ha tempo di vita $\tau+1$. Poiché le due τ non compaiono in una stessa frase, non dovrebbero dare adito a confusione.

$$\tau_0 = 1;\ \ \tau_{new} = \tau_{old} + 1 \tag{3.5}$$

2. Media non stazionaria, μ_{exp} Questa media è spesso sostituita alla media tradizionale, μ, in quanto pesa maggiormente termini più recenti e gradualmente dimentica quelli passati. Se si passa dalla definizione ricorsiva all'iterativa, ci si accorge che non è altro che una media pesata in cui il peso τ^{mo} è in buona misura un esponenziale funzione di α. In letteratura, viene chiamata *alpha-blending*, *exponential forgetting*, ... Questo parametro (insieme al successivo) vuole modellare una *distribuzione* Gaussiana **non stazionaria**. Nell'attuale implementazione, $\alpha = 0.9$ (come suggerito in [11]). Da notare che α misura l'adattatività e quindi la memoria del sistema. Nel caso limite $\alpha = 0$, la media è sempre l'ultimo valore (massima adattatività); nel caso opposto $\alpha = 1$, la media rimane ancorata al primo valore (massima memoria).

$$\mu_{exp_0} = x_0; \qquad \mu_{exp_{new}} = \alpha \cdot \mu_{exp_{old}} + (1 - \alpha)x_{new} \tag{3.6}$$

3. Sigma non stazionaria, σ_{exp} Discorso analogo al punto precedente, con la differenza che qui il valore tradizionale cui sostituirsi è la σ (ovvero lo scarto quadratico medio). La formula è stata presa a prestito da [11]. Anche qui si è posto $\alpha = 0.9$.

$$\begin{aligned} V_{exp_0} &= 0; \\ V_{exp_{new}} &= \alpha(V_{exp_{old}} + (\mu_{exp_{new}} - \mu_{exp_{old}})^2) + \\ &\quad +(1 - \alpha)(x_{new} - \mu_{exp_{new}})^2 \qquad (3.7) \\ \sigma_{exp} &:= \sqrt{V_{exp}} \qquad (3.8) \end{aligned}$$

4,5,6,7. Momenti di ordine $i(\in [1..4])$, M_i Essendo questo insieme la somma di tutti i valori che la variabile ha assunto dalla nascita (elevati alla potenza i^{ma}), è fondamentale per calcolare μ (vedi eq. (3.10)) e σ (vedi eq. (3.11)) della variabile. Questi due dati saranno molto utili se le x_i saranno supposte campioni di una *variabile aleatoria stazionaria*. Ovviamente non è necessario fermarsi a 4; questo numero pare però un buon

compromesso (vedi anche [6], pag. 508).

$$\begin{aligned} M_{i_0} &= x_0^i \\ M_{i_{new}} &= M_{i_{old}} + x_{new}^i \qquad (3.9) \\ \mu &= \frac{M_1}{\tau} \qquad (3.10) \\ \sigma &= \sqrt{\frac{M_2 - \mu^2}{\tau}} \qquad (3.11) \end{aligned}$$

8. Massimo scostamento, δ Questo è un dato certamente interessante che memorizza la massima variazione (in valore assoluto) tra due istanze successive del descrittore. Come la varianza, offre una misura di quanto il dato sia instabile, ma è ritenuto anche rappresentativo della rumorosità del segnale. É stato inserito in euristiche sull'affidabilità di un descrittore, ma a lavoro concluso è stato fondamentalmente tagliato fuori dai calcoli. Da notare che può essere calcolato grazie all'11^{ma} caratteristica (ℓ).

$$\begin{aligned} \delta_0 &= 0 \\ \delta_{new} &= \max\left\{ \delta_{old} , |\ell - x_{new}| \right\} \qquad (3.12) \end{aligned}$$

9. Valore massimo, x_{max} Molto semplicemente, il massimo valore assunto dalle x_i. É stato usato come ingrediente per la normalizzazione di σ (vedi Sez. 2.2.2) nel corso del lavoro ma alla fine è stato tolto.

$$\begin{aligned} x_{max_0} &= x_0 \\ x_{max_{new}} &= \max\left\{ x_{max_{old}} , x_{new} \right\} \qquad (3.13) \end{aligned}$$

10. Valore minimo, x_{min} É il minimo valore assunto dalle x_i. Non viene usato.

$$\begin{aligned} x_{min_0} &= x_0 \\ x_{min_{new}} &= \min\left\{ x_{min_{old}} , x_{new} \right\} \qquad (3.14) \end{aligned}$$

11. Ultimo valore, ℓ [5] É il valore assunto dal descrittore nell'ultima immagine. Si era ritenuto opportuno introdurre un piccolo elemento di memoria nella

[5] Buffamente, è anche l'ultima caratteristica :-)

statistica per poter fare dei confronti. Da questa idea è nato semplicemente (eq. (3.12)) δ, l'ottava caratteristica. Ha dunque più un uso interno che non un vero utilizzo nelle metriche utilizzate.

$$\ell_0 = x_0; \qquad \ell_{new} = x_{new} \tag{3.15}$$

3.3.2 Statistiche associate a un *lista* di descrittori

Ora che abbiamo definito la struttura dati associata a un singolo descrittore, il passo verso l'intero vettore di descrittori è breve.

Definizione 3.3.2 (History) *Data una sequenza $\mathcal{S}_\tau$, e una lista di λ descrittori $\boldsymbol{\varphi} := (\varphi_1, ..., \varphi_\lambda)$, si definisce* History $\boldsymbol{\mathcal{H}}$ *la matrice di dimensione* $(\lambda \times 11)$ *che ha per riga i^{ma} la* statistica *(che è un* vettore riga*) associata all' i^{mo} descrittore della lista.*[6]

$$\boldsymbol{\mathcal{H}}[\boldsymbol{\varphi}, \mathcal{S}_\tau] := \begin{bmatrix} \boldsymbol{\varsigma}(\mathcal{S}_\tau, \varphi_1) \\ \boldsymbol{\varsigma}(\mathcal{S}_\tau, \varphi_2) \\ \boldsymbol{\varsigma}(\mathcal{S}_\tau, \varphi_3) \\ ... \\ \boldsymbol{\varsigma}(\mathcal{S}_\tau, \varphi_\lambda) \end{bmatrix} \equiv \begin{bmatrix} \tau_1 & \mu_{exp_1} & \sigma_{exp_1} & \dots & \ell_1 \\ \tau_2 & \mu_{exp_2} & \sigma_{exp_2} & \dots & \ell_2 \\ \tau_3 & \mu_{exp_3} & \sigma_{exp_3} & \dots & \ell_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \tau_\lambda & \mu_{exp_\lambda} & \sigma_{exp_\lambda} & \dots & \ell_\lambda \end{bmatrix} \tag{3.16}$$

In futuro si farà largo uso di questa matrice dando però per scontata (in quanto fissa) la lista di descrittori $\boldsymbol{\varphi}$. Si tenderà perciò a usare la notazione semplificata $\boldsymbol{\mathcal{H}}[\mathcal{S}_\tau]$ (o $\boldsymbol{\mathcal{H}}_{\mathcal{S}_\tau}$) in vece di $\boldsymbol{\mathcal{H}}[\boldsymbol{\varphi}, \mathcal{S}_\tau]$.

3.3.3 Conclusioni

É stato creato un modello che associa a una sequenza di immagini λ vettori (uno per ogni descrittore) che rappresentino ciascuno una statistica su un aspetto diverso (grandezza, colore medio, omogeneità, ...) di un'immagine che varia nel tempo.

[6] Si noti che $\tau_1 = \tau_2 = \ldots = \tau_\lambda$, ovvero il tempo di vita di ogni descrittore è uguale poiché si comincia a calcolarli tutti alla nascita della sequenza. Si può dunque pensare ad un'imperfezione del modello oppure a un modello in grado di supportare descrittori che 'nascano' in momenti diversi - che era nelle intenzioni iniziali ma poi, per semplicità, si è persa nel corso dell'implementazione.

Il modello statistico che sta alla base è che ogni descrittore rappresenti nel tempo una variabile gaussiana (la cui stazionarietà o meno verrà decisa in fase di matching, vedi Cap. 4).

Ciò che preme sottolineare è che un insieme di N fenomeni gaussiani che *concorrono* a descrivere un fenomeno stocastico – già noto in letteratura come *Mixture of Gaussians (MOG)* – è di solito rappresentato con un vettore di N medie (nulla di nuovo) e da una *matrice* $(N \times N)$ *di covarianza*. La forte **ipotesi semplificativa** adottata in questo lavoro (e in [14], tanto per citarne uno) è stata considerare gli N fenomeni **scorrelati**; la matrice verrà dunque considerata diagonale ed ecco dunque valido il modello (si rammenta che per ognuno dei λ descrittori si ha *una* media e *una* varianza[7], non si ha un vettore di covarianze con gli altri descrittori!).

Questa semplificazione é stata ritenuta logicamente accettabile ed è certamente – da un punto di vista computazionale – molto vantaggiosa.

[7] replicate per il caso stazionario e non.

Capitolo 4

Metriche per il confronto di immagini e sequenze

In questo capitolo verranno descritti tutti i metodi utilizzati per *confrontare* più parti, ove ogni parte è *o* una singola immagine *o* una sequenza di immagini.

4.1 Tipi di distanza modellati e commenti

É importante puntualizzare fin da subito che la maggior parte delle metriche che verranno introdotte *non sono delle vere e proprie distanze*, nel senso proprio del termine. Con abuso di notazione, verranno chiamate tali ben sapendo che spesso non rispettano i requisiti di una distanza (proprietà triangolare mancata, distanze nulle tra vettori anche diversi, ...).

Il problema fondamentale di cui ci si è occupati è stato *decidere* se insiemi di immagini o sequenze rappresentassero una stessa cosa (i.e. sempre lo stesso ragazzo che cammina in due tempi diversi) o cose diverse (i.e. una persona e un furgone). Per fare questo è innanzitutto necessario *descrivere* ogni oggetto e per questo si sono spese già parole, modelli e calcoli nei capitoli precedenti (Cap. 2 e Cap. 3). Una volta descritta un'immagine o una sequenza di queste (nel secondo caso ci saranno sicuramente molti più dati da usare), quel che avremo in mano sarà un insieme di dati ben più piccolo dell'originale ma (possibilmente) quasi altrettanto rappresentativo. Per intenderci, non avremo più la bitmap di partenza

ma una cinquantina di numeri che ne descrivono la grandezza, il contrasto, il tono di grigio più frequente, ...

Importante notare da subito che questo modello *già* impone dei vincoli: se si vogliono confrontare due immagini, per esempio, non si può inventare una distanza che le confronti pixel per pixel, poiché ciò che si ha in mano di ciascuna di esse è un piccolo sottoinsieme dell'informazione originaria. Tutto ciò che si può fare è sfruttare queste poche informazioni per definire opportune funzioni di matching.

Principio 4.1.1 (Ipotesi di 'passaggio obbligato' *per* i descrittori) *Ogni metrica (funzione di* matching*) non può conoscere di un'immagine* $\mathcal{I}$ altro *che il suo* **vettore di Features** $\boldsymbol{v}[\mathcal{I}]$ *(Def. (3.2.2), pag. 38). Analogamente, ogni sequenza* $\mathcal{S}_\tau$ *coinvolta in una metrica contribuirà solo con i parametri contenuti nella sua* **history** $\boldsymbol{\mathcal{H}}[\boldsymbol{\varphi}, \mathcal{S}_\tau]$ *(Def. (3.3.2), pag. 44).* [1]

$$d(\mathcal{I}_1, ..., \mathcal{I}_m, \mathcal{S}_1, .., \mathcal{S}_n) \text{ ammissibile se :} \qquad (4.1)$$
$$\exists\ \tilde{d}\Big(\boldsymbol{v}[\mathcal{I}_1], ..., \boldsymbol{v}[\mathcal{I}_m], \boldsymbol{\mathcal{H}}_{\boldsymbol{\varphi}}[\mathcal{S}_1], ..., \boldsymbol{\mathcal{H}}_{\boldsymbol{\varphi}}[\mathcal{S}_n]\Big)$$

I descrittori, insomma, sono funzioni *unarie* di un'immagine (o sequenza) e la più generica funzione che possa essere calcolata per confrontarne due è una funzione che abbia per argomenti i λ descrittori di ciascuna immagine. Si può confrontarne l'altezza, la forma, il colore più frequente ma non il secondo pixel in alto a destra (a meno che non rientri nella rosa di descrittori scelta).

Chiarito questo semplice concetto a livello d'*interfaccia* (che però già impedisce di usare alcune delle metriche più famose in letteratura, come i confronti tra istogrammi)[2], è opportuno parlare di un'importante scelta progettuale che ha permeato *tutte* le distanze create:

Principio 4.1.2 (Ipotesi d'indipendenza dei descrittori) *Ci si disinteressa completamente della semantica di ogni descrittore, considerandoli tutti indipendenti e scorrelati tra loro; si assume che possano avere 'unità di misura' diverse, escludendo quindi di poterli (in generale) confrontare tra loro. Non è*

[1] L'informazione si ridurrà, per ciascun descrittore, a un solo numero per un'*immagine* o a 11 numeri per una *sequenza*.

[2] a meno che, ovviamente, non si prendano *tutti* i dati dell'istogramma come descrittori.

*garantita l'*effettiva scorrelazione *tra di essi*[3]*, ma questa viene supposta al fine dei calcoli.*

Questa scelta è molto forte e impedisce, per esempio, di trascurare (in una certa distanza) tutte le caratteristiche di tessitura se l'immagine è sufficientemente piccola (o allungata, o scura...). L'unica eccezione a questo fatto è il tempo di vita (eq. (3.5), pag. 41) di una sequenza che viene effettivamente usato; c'è però da dire che questo non è un vero e proprio parametro di un descrittore, in quanto è comune a tutti ed è quindi piuttosto una caratteristica della sequenza stessa.

Tutte le metriche tentate, dunque, tendono a confrontare – tra due oggetti – i loro λ descrittori uno ad uno, e si caricheranno di λ piccoli contributi, ciascuno funzione dei due *vettori di features* associati al passo i^{mo}.

Questo significa che, per confrontare due immagini, non si può (per come è stato impostato il lavoro) inventare una generica funzione $f(\mathcal{I}_1, \mathcal{I}_2)$, ma occorre passare per una *meno generica* funzione $f'(\varphi_k(\mathcal{I}_1), \varphi_k(\mathcal{I}_2))$.

Un'ultima considerazione importante: si parlerà spesso, più avanti, di metriche **stazionarie** vs **non stazionarie**. Questo ha senso solo quando *almeno* un oggetto contemplato è una sequenza. In pratica molti calcoli sono imperniati su **valore medio** (μ) e **scostamento quadratico medio** (σ) di un certo descrittore. Molto semplicemente, la scelta della stazionarietà influisce sulla scelta di μ e σ (e varierà dunque completamente i risultati):

- nella metrica *stazionaria* i due dati verranno estratti dai Momenti (quindi dalle features numero 1,4 e 5 della *statistica* $\boldsymbol{\varsigma}(\mathcal{S}_\tau, \varphi_n)$), secondo le equazioni (eq. (3.10), pag. 43) e (eq. (3.11), pag. 43) rispettivamente;

- nella metrica *non stazionaria* verranno copiati brutalmente dalle features 2 e 3 di $\boldsymbol{\varsigma}(\mathcal{S}_\tau, \varphi_n)$, secondo le equazioni (eq. (3.6), pag. 42) e (eq. (3.8), pag. 42) rispettivamente.

Nei paragrafi che seguono verranno descritti tipi diversi di distanze in ordine di complessità (e quindi in ordine *inverso* di generalità). É anche l'ordine cronologico che ha seguito l'effettiva implementazione.

[3]potrei avere due entropie che sono *sempre* l'una il doppio dell'altra.

4.2 Dati associati a Immagini e Sequenze

Prima di descrivere le funzioni di matching vere e proprie, si ritiene opportuno rispolverare i dati che sono disponibili quando ci si appresta a confrontare tra loro Immagini o Sequenze.

- Data un'*immagine* $\mathcal{I}_{[X,Y]}$, tutto ciò che si ha è la rosa di descrittori applicati ad essa; dunque il **vettore di Features** (Def. (3.2.2), pag. 38) $\boldsymbol{v}[\mathcal{I}] \equiv (\varphi_1(\mathcal{I}), ..., \varphi_\lambda(\mathcal{I}))$.
- Data una *sequenza* $\mathcal{S}_\tau$, si ha molto di più: ognuno dei λ descrittori ha una sua storia, rappresentata dalla sua *statistica* (Def. (3.3.1), pag. 40) $\boldsymbol{\varsigma}(\mathcal{S}_\tau, \varphi_i)$. Si avranno quindi λ statistiche che andranno a formare la History (Def. (3.3.2), pag. 44) della sequenza, che prende la forma della matrice $\boldsymbol{\mathcal{H}}[\boldsymbol{\varphi}, \mathcal{S}_\tau]$ (vedi Eq. (3.16)).

Terminato questo ripasso, possiamo procedere con le distanze vere e proprie.

4.3 Distanza tra due Immagini $\mathcal{I}_A$ e $\mathcal{I}_B$

La più semplice delle distanze è quella tra due immagini. Tutto quello che si ha in mano sono 2λ parametri confrontabili a due a due. É importante sottolineare che, non conoscendo la scala dei valori, occorre fare in modo che diano contributi simili ed evitare che un valore superiore ad altri di diversi ordini di grandezza alteri completamente il risultato della distanza. Urge dunque una *normalizzazione* di ogni descrittore quando non è disponibile nemmeno una statistica cui rifarsi!

In generale tutte le distanze di questo tipo seguono il seguente *pattern*:

$$d(\mathcal{I}_A, \mathcal{I}_B) = \sum_{i=1}^{\lambda} \delta(\varphi_i(\mathcal{I}_A), \varphi_i(\mathcal{I}_B)) \tag{4.2}$$

Questo approccio è necessario se si vuole rispettare l'*ipotesi d'indipendenza dei descrittori* (Princ. (4.1.2), pag. 48).

Poiché qui non v'è alcun elemento statistico su cui basarsi, le uniche distanze adottate sono le tre più famose: $||\cdot||_1$, $||\cdot||_2$ (norma 'euclidea') e $||\cdot||_\infty$ (norma del *sup*). Per comodità di notazione si supporrà:

$$a_i := \varphi_i(\mathcal{I}_A); \qquad b_i := \varphi_i(\mathcal{I}_B);$$

Le distanze saranno dunque: [4]

$$d_1(\mathcal{I}_A, \mathcal{I}_B) = \sum_{i=1}^{\lambda} \frac{1}{\lambda} \left| \frac{b_i - a_i}{b_i + a_i} \right| \tag{4.4}$$

$$d_2(\mathcal{I}_A, \mathcal{I}_B) = \sum_{i=1}^{\lambda} \frac{1}{\lambda} \left(\frac{b_i - a_i}{b_i + a_i} \right)^2 \tag{4.5}$$

$$d_\infty(\mathcal{I}_A, \mathcal{I}_B) = \max_{i \in N_\lambda} \left(\left| \frac{b_i - a_i}{b_i + a_i} \right| \right) \tag{4.6}$$

Si noti che queste distanze sono state normalizzate a valori in $[0, 1]$. Queste distanze non sono praticamente state usate nel lavoro svolto.

4.4 Distanza tra un'immagine $\mathcal{I}_{target}$ e una sequenza $\mathcal{S}_\tau$

Questa volta il discorso si fa molto interessante, sia da un lato concettuale che pratico: il problema consiste nel risolvere la questione *"quanto un'immagine $\mathcal{I}_{target}$ assomiglia all'insieme di immagini dato"*? Potremmo avere una sequenza e chiederci in quale misura un'immagine possa essere considerata istanza successiva della sequenza stessa (nel qual caso verranno probabilmente usate stime non stazionarie, poiché nel calcolo saranno ritenute più significative le ultime immagini della sequenza, rispetto alle meno recenti).

Qui vi sono molte più informazioni, poiché le λ statistiche associate a $\mathcal{S}_\tau$ permettono di fare un matching *ad hoc* per l'immagine. Per ogni descrittore, infatti, vi sono indicazioni sulla sua stabilità nel tempo. Se si considera il suo variare nel tempo come una variabile aleatoria (e si confondono *frequenze* con *probabilità*, come spesso si fa in statistica), una buona misura della stabilità viene fuori da σ. Purtroppo, ancora una volta, non possono essere confrontate tra loro le $\sigma_{1..\lambda}$, poiché variano su range diversi: bisogna normalizzarle. Come già motivato in Sez. 2.2.2 (pag. 14), si è scelto come σ *normalizzato* $\overline{\sigma} := \frac{\sigma}{\mu}$. É una soluzione molto semplice ed effettivamente funziona.

[4] se il denominatore è nullo per un certo i, il contributo i^{mo} alla somma sarà nullo. Questa filosofia (a volte giustificata a volte meno) rimarrà valida in tutti i calcoli futuri per praticità.

Qui il modello generale, dunque, si complica (posto $\varphi_i(\mathcal{I}) =: x_i$):

$$d(\mathcal{I}, \mathcal{S}_\tau) = \frac{\sum_{i=1}^{\lambda} w_i(\boldsymbol{\varsigma_i}(\mathcal{S}_\tau)) \cdot \delta_i(x_i, \boldsymbol{\varsigma_i}(\mathcal{S}_\tau))}{\sum_{i=1}^{\lambda} w_i(\boldsymbol{\varsigma_i}(\mathcal{S}_\tau))}, \tag{4.7}$$

dove w_i sta per peso i^{mo} della sommatoria ed è un indice di affidabilità della misura, $\delta(x_i, \boldsymbol{\varsigma_i}(\mathcal{S}_\tau))$ è la *distanza parziale* i^{ma}, mentre $\boldsymbol{\varsigma}_i(\mathcal{S}_\tau)$ è la statistica associata alla sequenza per il descrittore i^{mo}. Questo modello generale si specializza (nel presente lavoro) in un modello che sfrutti semplicemente media e varianza del descrittore; allora si avrà:

$$d(\mathcal{I}, \mathcal{S}_\tau) = \frac{\sum_{i=1}^{\lambda} w_i(\bar{\sigma}_i, x_i, \mu_i) \cdot \delta_i(x_i, \mu_i)}{\sum_{i=1}^{\lambda} w_i(\bar{\sigma}_i, x_i, \mu_i)}, \tag{4.8}$$

dove questa distanza sortirà due risultati diversi nei casi stazionario e non (perché sarà diverso il modo di calcolare μ e σ, vedi Sez. 4.1).

Questo vuole effettivamente dire che molti parametri interni alla *statistica* non sono stati usati. In realtà, si è tentato con euristiche che influissero sulle affidabilità parziali (cioé i w_i) tramite l'età, il range dinamico (max e min) e altre cose, ma non si sono mai riscontrati apprezzabili miglioramenti. Questo vuol dire che in futuro si potrà troncare drasticamente l'occupazione del vettore $\boldsymbol{\varsigma_i}$.

Nota 4.4.1 *Non vi è, per come è stato affrontato il lavoro, un'idea di* ***bontà della distanza****. É interessante almeno accennare all'interesse pratico che rivestirebbe: se si volesse incapsulare nella distanza questo fattore, si potrebbe in futuro pensare di calcolare più distanze (derivanti da oggetti magari incommensurabili tra loro) e sapere quale è più attendibile tra di esse. Questo è particolarmente importante in (Sez. 4.7), dove ci sono tanti numeri ma li si può (specialmente tra righe diverse) confrontare tra loro. Un possibile (e semplice) sviluppo futuro è estendere ogni distanza come funzione a due valori in uscita:* valore vero e proprio *della distanza e attendibilità della stessa.*

4.4.1 Distanza pseudo-euclidea tra $\mathcal{I}_{target}$ e $\mathcal{S}_\tau$

La prima distanza che viene in mente è l'euclidea. Nella versione attuale (che cambia a seconda che sia stazionaria o meno), è stata introdotta inoltre qualche piccola euristica sulla stabilità dei descrittori.

Seguono la definizione di distanza parziale e di peso parziale secondo il modello (eq. (4.8)) nei casi stazionario e non.

$$\delta_i = 100 \cdot \left| \frac{x_i - \mu_i}{x_i + \mu_i} \right|^K \tag{4.9}$$

$$w_{staz._i} = \begin{cases} 0, & \bar{\sigma} \geq 0,2; \\ 1 - \bar{\sigma}_i, & 0,1 < \bar{\sigma}_i < 0,2; \\ 3(1 - \bar{\sigma}_i), & \bar{\sigma}_i < 0,1; \end{cases} \tag{4.10}$$

$$w_{nonstaz._i} = \frac{1}{1 + \bar{\sigma}_i^2} \tag{4.11}$$

Note:

- Si è provato con K=2 (una distanza, quindi, somigliante all'euclidea) ma per K=1 la funzione va un po' meglio (strano, poiché in teoria il quadrato dovrebbe rimpicciolire valori sotto a 1 e amplificare quelli superiori – quindi fondamentalmente aumentare il potere discriminante tra cose vicine e lontane).

- Il valore a soglia dei pesi stazionari è dettato da un'euristica molto semplice: *non contribuiscono affatto alla misura* quei descrittori troppo altalenanti (più del 20%), mentre sono premiati i più stabili (meno del 10%) triplicandone il peso.

- Il peso $w_{nonstaz.}$ è calcolato diversamente poiché si è osservato sperimentalmente un range molto più 'esteso' per i campioni σ_{exp}. Per intenderci, mentre i $\bar{\sigma}$ stazionari eccedono *molto raramente* l'unità (100%), sono stati trovati più spesso valori sballati per i $\bar{\sigma}_{exp}$. Che sia per errori implementativi o perché il parametro è più instabile al momento non si sa, fatto sta che

si è optato per questa seconda equazione[5] (eq. (4.11)) come proposto anche in ([6], pag. 508). Ci si potrebbe chiedere perché, viceversa, non sia stata adottata *sempre* questa stima. Ebbene, il fatto è che la curva $\frac{1}{1+x^2}$ per x prossime a zero (nel nostro caso, $\bar{\sigma}$ basse quindi campioni buoni) ha una bassissima sensibilità (derivata pressoché nulla) e quindi non si può discriminare tra campioni molto stabili. Anzi, si ritiene che una buona funzione di peso dovrebbe premiare *esponenzialmente* con la vicinanza di $\bar{\sigma}$ a 0, ma non ci si è addentrati ancora in tal direzione (già la triplicazione nel caso non-stazionario è stata un *azzardo*).[6]

4.4.2 Distanza a soglia a valor fisso

Verrà qui presentata una metrica molto semplice che però ha dato grosse soddisfazioni.

L'idea consiste nel sommare per ogni descrittore un valore secondo una soglia: se il match i^{mo} è buono sommiamo 1, se no sommiamo 0. La distanza si riduce alla 'percentuale' di match buoni, se alla fine il risultato viene normalizzato. Anche qui ci sarà una distanza stazionaria e una non, a seconda di come calcolo μ e σ.

$$\delta_i := \begin{cases} 0, & |\mu_i - x_i| \leq k\bar{\sigma}; \\ 1, & |\mu_i - x_i| > k\bar{\sigma}; \end{cases} \tag{4.12}$$

$$w_i = 1 \tag{4.13}$$

Nell'implementazione attuale, si è ritenuto $k = 3$ essere un buon compromesso (se k è troppo alto, tutto sembra uguale, se è troppo basso paiono diverse anche cose simili). La formula trova la sua giustificazione teorica nel considerare l'i^{mo} descrittore una variabile aleatoria *gaussiana* (stazionaria o meno); questa distanza indica una stima semplice di probabilità che l'immagine appartenga alla

[5] É pur sempre una affidabilità in funzione dello scostamento normalizzato, che tende a uno per $\bar{\sigma}$ bassi e tende a 0 per $\bar{\sigma}$ alti.

[6] quando vi sono 50 descrittori da far pesare in una distanza, la paura è di eccedere con un'euristica e arrivare al punto in cui uno di questi 'copre' gli altri (magari perché ha un peso 100 volte maggiore), e potrebbe farlo solo perché per lui il parametro non è significativo e quindi costante. Il malfunzionamento sarebbe peraltro poco *osservabile*. Nel dubbio i pesi sono stati tenuti abbastanza omogenei...

classe (nella *forte* ipotesi che i descrittori siano scorrelati tra loro, ovvero che la matrice di covarianza sia diagonale) . Modelli più complessi possono senz'altro avere un migliore supporto teorico, ma già questa funzione dà buoni risultati ed è implementativamente semplicissima (e ben si presta a piccole modifiche, come per esempio la distanza a pag. 55).

Analizzando la formula, ci si rende conto che *non* è una distanza: può essere nulla per immagini diverse (purché entro 3σ per ogni descrittore), non è simmetrica (accetta argomenti diversi!) e non ha senso parlare di disuguaglianza triangolare (ricordiamo che $d(\mathcal{I}_1, \mathcal{I}_2) \leq d(\mathcal{I}_1, \mathcal{S}_\tau) + d(\mathcal{I}_2, \mathcal{S}_\tau)$ non ha senso poiché una delle tre parti vorrebbe due immagini, il che è incompatibile con la definizione).[7]

4.4.3 Distanza a soglia a valore variabile

Non è altro che una variante della distanza del paragrafo precedente. Non fa altro che penalizzare valori che si allontanino di più dal valore di soglia (mentre nella eq. (4.12) la distanza è sempre uguale sia a $4\,\bar{\sigma}$ che a $42\,\bar{\sigma}$, tanto per dire). Stranamente, da un punto di vista sperimentale, la distanza a pag. 54 rimane la più soddisfacente. La distanza è definita come:

$$\text{Sia } \epsilon := |\mu_i - x_i| - k\sigma; \tag{4.14}$$

$$\delta_i = \begin{cases} f(\epsilon), & \epsilon > 0; \\ 0, & \epsilon \leq 0; \end{cases} \tag{4.15}$$

$$w_i = 1, \tag{4.16}$$

dove $f(x)$ una qualunque funzione ritenuta opportuna; come argomento essa ha l'*eccedenza* ϵ (sempre positiva, altrimenti δ_i varrebbe 0), ovvero di quanto μ_i esce dal range prestabilito. Questa eccedenza può essere normalizzata a $\bar{\epsilon} := \frac{\epsilon}{\max(\mu_1, \mu_2)}$, al fine di garantire di non premiare descrittori a valori 'grandi'. Le funzioni

[7] e se volessimo fingere che fosse possibile, esisterebbero $\mathcal{I}_1, \mathcal{I}_2, \mathcal{S}_\tau$ che la violino: basti pensare a due immagini a 2σ dalla sequenza (per ogni descrittore), ma a 4σ tra loro. Avremmo dunque $d(> 0) \leq 0 + 0$...

provate sono:

$$f_1(x) = log(\epsilon) \tag{4.17}$$

$$f_2(x) = \bar{\epsilon}^2 \tag{4.18}$$

$$f_3(x) = \bar{\epsilon} \tag{4.19}$$

1. L'idea di sommare il logaritmo di quanto l'immagine *deborda* dalla statistica nasce dall'esigenza pratica di attutire valori anche molto dissimili in quantità molto vicine tra loro. Sebbene abbia un basso supporto teorico[8], è una soluzione molto funzionale.

   La letteratura è piena, tra l'altro, di conversioni logaritmiche per indici di probabilità, per la proprietà fondamentale che 'tramuta' somme in prodotti (e il prodotto ha il significato logico dell'AND, ben più pregnante di una somma).[9]

2. La funzione f_2 non fa altro che sommare il quadrato dell'eccedenza normalizzata (qualora ci sia *effettivamente* un'eccedenza) per ogni descrittore; non è altro che una variante della distanza euclidea.

3. Analogamente a f_2, la funzione f_3 esegue la somma *semplice* delle eccedenze normalizzate; può dunque sembrare una variante della distanza in $||\ \cdot\ ||_1$. Nella versione corrente, questa *è la versione usata.*

4.4.4 Distanza retroazionata (cenni)

Questa è un'idea allo stato embrionale che non è stata ancora realizzata ma viene qui citata per il suo interessante. Più che una distanza, è un'idea di *variante* applicabile a qualunque distanza di tipo $d(\mathcal{I}, \mathcal{S}_\tau)$ non stazionaria.

L'idea di fondo è: occorre calcolare la distanza tra un'immagine $\mathcal{I}_{target}$ e una sequenza $\mathcal{S}_\tau$. Anziché applicare brutalmente $d(\mathcal{I}, \mathcal{S}_\tau)$, si calcola una distanza

[8] avrebbe avuto piuttosto senso sommare il valore stesso (poiché già ha una distribuzione gaussiana quindi è sempre più improbabile che si distacchi di $N\bar{\sigma}$ con legge – come noto – esponenziale).

[9] se la distanza parziale è giustificata come probabilità di un evento singolo, la somma delle distanze parziali *in forma logaritmica* può trovare giustificazione come probabilità riferita al problema globale. Per un esempio, vedere [15].

esterna (quella che avrebbe dovuto essere la distanza vera e propria) e una distanza interna (tra la sequenza e la *sua* ultima immagine). Sia $\mathcal{I}_{last}$ l'ultima immagine di $\mathcal{S}_\tau$. Avremo:

$$\begin{cases} d_{int} & := & d(\mathcal{I}_{last}, \mathcal{S}_\tau) \\ d_{ext} & := & d(\mathcal{I}_{target}, \mathcal{S}_\tau) \end{cases} \tag{4.20}$$

$$d'(\mathcal{I}_{target}, \mathcal{S}_\tau) = f(d_{int}, d_{ext}) \tag{4.21}$$

Questo dà una specie di *contesto* in cui muoversi e dà un'idea esterna di *attendibilità* (esterna poiché non occorre creare una funzione che entri dentro alla statistica di un descrittore, si chiama semplicemente una distanza già fatta e la si confronta con un secondo dato): si può penalizzare un calcolo perché funziona *male* sulla sua stessa sequenza o premiarlo se va bene. Si potrebbe complicare il modello facendo entrare f a livello di singolo descrittore (quindi penalizzare un descrittore che classifica in modo errato l'ultima immagine della propria sequenza in favore di uno che ci prende) rendendolo – se non più giusto – più giustificato.

Questa rimane comunque un'idea cui non possono ancora essere affiancati dei risultati; e tutto il discorso crolla (da un punto di vista concettuale) nel caso stazionario.[10]

4.5 Distanza tra due sequenze

4.5.1 Introduzione

É molto interessante chiedersi se una sequenza ha un buon *matching* con un'altra in applicazioni di Tracking. Un problema comune è continuare a inseguire più oggetti in movimento anche quando questi scompaiano di scena (o dietro a un elemento fisso *in* scena), o si occludano a vicenda (anche solo parzialmente). Purtroppo il modulo di estrazione dei blob (visti come macchie connesse) non sa giudicare se questo contenga uno o più oggetti (visti come unità *interessanti*, quindi unità logiche e non fisiche). Avendo a disposizione una metrica *perfetta* di tipo $\mathrm{d}(\mathcal{I}, \mathcal{S}_\tau)$, ci si potrebbe aspettare di aver risolto *tutti* i problemi di Tracking, e ciò è in buona misura vero: appena l'elemento da studiare ritorna visibile (rientra in scena o si stacca dagli altri), al primo fotogramma disponibile si decide se

[10] Perché dovrei mai usare l'ultima immagine e non le altre? Sono tutte paritarie tra loro!

appartiene o meno a sequenze precedentemente calcolate (diremo *congelate*).[11] Poiché, però, in casi reali questo nuovo fotogramma non dà un risultato soddisfacente (o perché è rumoroso o perché la metrica non è perfetta e non offre una risposta decisa), il nostro gruppo di lavoro ha deciso di adottare la strategia "**freeze-and-wait**", ovvero:

Freeze Quando un oggetto inseguito non è più *tracciabile* (o perché esce di scena o perché c'è un **merge** con altri oggetti) congeliamo la conoscenza acquisita (in generale la sua *storia*, ovvero i dati che riteniamo significativi per tracciarlo).

Wait Quando l'oggetto inseguito ritorna *trackabile* (o perché rientra in scena o perché c'è uno **split** dal gruppo formato con altri oggetti), riprendiamo in mano i dati conosciuti e cerchiamo di scoprire *chi sia* quest'oggetto appena nato: è un oggetto nuovo, o assomiglia a uno di quelli entrati nel gruppo? Potremmo dare una *risposta immediata* (il che è auspicabile, poiché poi quest'oggetto può entrare dopo pochi frame in un nuovo gruppo) o possiamo scegliere di *aspettare*, accumulando informazioni finché dopo qualche fotogramma la nostra metrica non sia in grado di darci una risposta decisa (che *non* vuol dire corretta).

Allora è facile dedurre l'importanza teorica di una funzione di matching tra due sequenze: si ha a disposizione una sequenza $\mathcal{S}_\tau^{old}$ fissa (versione 'vecchia' di un oggetto) e una sequenza *in fieri* che nasce come immagine singola $\mathcal{S}_0^{new}$, poi frame dopo frame diventa una sequenza sempre più consistente (da un punto di vista statistico): $\mathcal{S}_1^{new}, \mathcal{S}_2^{new}, ..., \mathcal{S}_t^{new}$.

Nota. Si noti che il **freeze n' wait** non è l'unico approccio possibile: quando perdiamo visibilità di un oggetto smettiamo di fare calcoli (per quell'oggetto) finché non torni disponibile nella sua interezza. Altri ricercatori [12], invece, continuano a inferire informazioni su di esso anche durante l'occlusione, stabilendo –

[11] In realtà un modello reale ha molti problemi che nemmeno una funzione di matching perfetta può risolvere (il che lascia ben sperare) a meno che non si introducano delle conoscenze sul mondo da studiare (concetto di persona, automobile, punti d'ingresso della scena, ...). Una persona, a causa del rumore, può decomporsi in 3 parti e ricomporsi, e nessuna macchina a questo livello potrà sapere se è un oggetto o sono tre.

in un gruppo di oggetti – chi sia (per esempio) davanti e chi dietro. Quello usato è tuttavia un approccio più che sensato; essendo più restrittivo, potrebbe essere però necessario in futuro aprirsi a modelli più generali.

4.5.2 Distanze provate

Un modello generale può essere:

$$d(\mathcal{S}_A, \mathcal{S}_B) = \frac{\sum_{i=1}^{\lambda} w_i(\varsigma_{\boldsymbol{i}}(\mathcal{S}_A), \varsigma_{\boldsymbol{i}}(\mathcal{S}_B)) \cdot \delta_i(\varsigma_{\boldsymbol{i}}(\mathcal{S}_A), \varsigma_{\boldsymbol{i}}(\mathcal{S}_B))}{\sum_{i=1}^{\lambda} w_i(\varsigma_{\boldsymbol{i}}(\mathcal{S}_A), \varsigma_{\boldsymbol{i}}(\mathcal{S}_B))}, \tag{4.22}$$

Sono stati aggiunti dei nomi di comodo in modo da potersi riferire meglio alle distanze che seguono nel Cap. 5.

Distanza SS_1 (semplice) Un caso abbastanza semplice che è stato provato è il seguente[12]:

$$w_i(\varsigma_{\boldsymbol{i}}(\mathcal{S}_A), \varsigma_{\boldsymbol{i}}(\mathcal{S}_B)) = \begin{cases} 1, & se\ (\bar{\sigma_1} \leq 5) \wedge (\bar{\sigma_2} \leq 5) \\ 0, & altrimenti. \end{cases} \tag{4.23}$$

$$\delta_i(\varsigma_{\boldsymbol{i}}(\mathcal{S}_A), \varsigma_{\boldsymbol{i}}(\mathcal{S}_B)) = \frac{|\mu_1 - \mu_2|}{\frac{\sigma_1 + \sigma_2}{2}} \tag{4.24}$$

Quindi si fa per ogni descrittore il seguente discorso: se una delle due statistiche è troppo 'sballata', non lo si fa incidere nel calcolo (5 *è* un valore altissimo, ma non si voleva rischiare un denominatore nullo; d'altronde il discorso funziona anche se non viene scartato alcun valore!); altrimenti tale descrittore interviene con un contributo pari alla differenza tra i valori medi normalizzata attraverso un numero che è funzione delle due varianze, che nell'attuale implementazione è la semplice media tra le σ.

Sarebbe molto significativo porre al denominatore della eq. (4.24) anziché la media tra le σ il valore minimo[13]; se infatti una *statistica* è molto 'convinta'

[12]Da notare una pecca formale: non è garantito che il denominatore sia maggiore di zero. In questo caso sarà usata una distanza arbitrariamente grande. C'è da dire che però non è mai capitata una situazione del genere.

[13]che vuol dire: "prendi la massima tra le due distanze viste da ogni sequenza".

di un valore (basso σ) e l'altra no, si può pensare di tenere la distanza più grande tra le due. L'attuale è stata una scelta di comodo poiché purtroppo accade spesso vi siano σ nulle (per esempio, ogni volta che il tempo di vita è unitario) e avere un valore diverso da zero *inquadra* correttamente il calcolo (poiché restituisce un ordine di grandezza che si può usare – a differenza dello zero).

Distanza SS_2 (simmetrica) Poiché questa distanza non dava risultati convincenti si è pensato di poggiarne una sul lavoro già fatto e ricondursi a una distanza del tipo Sequenza - Immagine (eq. (4.7), pag. 52). Date una generica distanza $d_{I,S}(\,\cdot\,,\,\cdot\,)$, due sequenze $(\mathcal{S}_A;\mathcal{S}_B)$ e dette $(\mathcal{I}^A_{last};\mathcal{I}^B_{last})$ (rispettivamente) la loro ultima immagine, possiamo definire ingenuamente una distanza come:

$$d'_{S,S}[d_{I,S}] := \frac{d_{I,S}(\mathcal{I}^B_{last},\mathcal{S}_A) + d_{I,S}(\mathcal{I}^A_{last},\mathcal{S}_B)}{2} \tag{4.25}$$

Non che sia una cattiva definizione (dipende ovviamente da quanto sia buona $d_{I,S}$!), però si può fare di meglio: considerare ogni sequenza come un'immagine singola prendendone i valori medi nel tempo (ancora rimane da decidere se stazionari o meno) e fingendo che sia un'immagine per cui i descrittori calcolino, volta per volta, un valore pari alla sua μ_i. Possiamo allora semplicemente definire:

Definizione 4.5.1 (Immagine Virtuale Di Una Sequenza) *Data una* history $\boldsymbol{\mathcal{H}}[\boldsymbol{\varphi},\mathcal{S}_\tau]$,[14] *si definisce* Immagine Virtuale Di Una Sequenza *un'ipotetica immagine $\mathcal{I}_\mathcal{V}(\mathcal{S}_\tau)$ per cui valga*[15]*:*

$$\forall i \in [1..\lambda] : \qquad \Big(\varphi_i(\mathcal{I}_\mathcal{V}) = \mu_i(\mathcal{S}_\tau)\Big) \tag{4.26}$$

Ancora una volta, si avranno due immagini virtuali per ogni sequenza a seconda che si usi una metrica stazionaria o meno. Nel primo caso si deriverà μ_i dalla (eq. (3.10), pag. 43), nell'altro dalla (eq. (3.6), pag. 42).

[14] costruita a partire dalla sequenza $\mathcal{S}_\tau$ e dal vettore di descrittori $\boldsymbol{\varphi}$, vedi (eq. (3.16), pag. 44)
[15] per definizione, non importa che detta immagine venga effettivamente trovata.

La distanza diventa allora:

$$d''_{S,S}[d_{I,S}] := \frac{d_{I,S}(\mathcal{I}^B_{\mathcal{V}}, \mathcal{S}_A) + d_{I,S}(\mathcal{I}^A_{\mathcal{V}}, \mathcal{S}_B)}{2} \tag{4.27}$$

ed è molto più significativa. Può essere semplicemente interpretata come la media tra le distanze tra due sequenze in cui volta per volta consideriamo fissa l'una – interpretandola come immagine – e mobile (nel senso di 'carica delle sue proprietà temporali') l'altra. Nell'implementazione attuale, si è usata (come distanza di partenza) la distanza a soglia intera (Sez. 4.4.2, pag. 54).

Distanza SS_3 (pesata) In questa distanza sono stati incrociati due fattori di peso: ogni descrittore pesa differentemente secondo la sua stabilità ($\hookrightarrow \sigma$); inoltre ciascuna delle due sequenze pesa differentemente secondo la propria età ($\hookrightarrow \tau$): è un calcolo apparentemente simmetrico in cui la sequenza più anziana pesa maggiormente nelle decisioni coi propri parametri.

$$w_i(\boldsymbol{\varsigma_i}(\mathcal{S}_A), \boldsymbol{\varsigma_i}(\mathcal{S}_B)) = \frac{\tau_1}{\bar{\sigma}_1} + \frac{\tau_2}{\bar{\sigma}_2} \tag{4.28}$$

$$\delta_i(\boldsymbol{\varsigma_i}(\mathcal{S}_A), \boldsymbol{\varsigma_i}(\mathcal{S}_B)) = \frac{|\mu_1 - \mu_2|}{\max(\mu_1, \mu_2)} \tag{4.29}$$

4.6 Distanza di un'Immagine $\mathcal{I}$ da *due* sequenze.

Questo tipo di distanza (apparentemente macchinoso e inutile) ben si presta a risolvere una semplice classe di problemi. Supponiamo di avere le sequenze $\mathcal{S}_A$ e $\mathcal{S}_B$ (vedere (fig. (1.1), pag. 5)) di due oggetti che sono stati inseguiti fino ad un certo istante t_1; da quel momento i due oggetti si sono fusi in un'unica componente connessa. In t_2, si separano di nuovo in due oggetti. Il problema di tracking che ci si trova a dover risolvere è: *"chi è chi?"*. Si potrebbe pensare inizialmente a una regola di decisione $f(\mathcal{S}_A, \mathcal{S}_B, \mathcal{I}_C, \mathcal{I}_D)$ che dia come risposta *booleana*: **0** – se $\mathcal{I}_C$ corrisponde a $\mathcal{S}_A$ e $\mathcal{I}_D$ corrisponde a $\mathcal{S}_B$; **1**, nel caso opposto ($\mathcal{I}_C$ corrisponde a $\mathcal{S}_B$ e $\mathcal{I}_D$ corrisponde a $\mathcal{S}_A$).

Ci accorgiamo subito però che questa definizione è un po' ingenua: nulla garantisce che i due oggetti uscenti dallo *split* siano effettivamente gli originali;

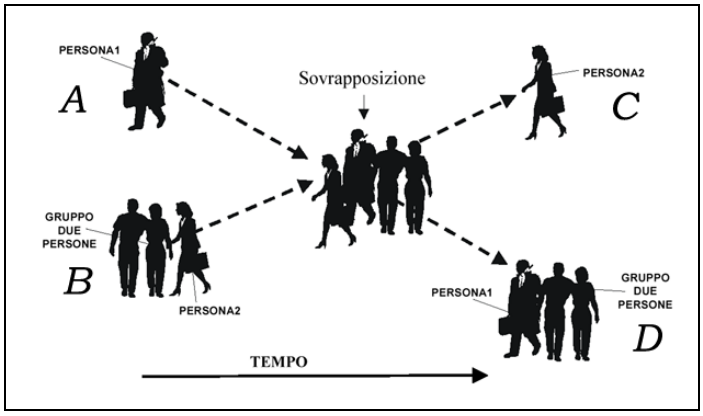

Figura 4.1: Esempio di collisione tra blob con rimescolamento dei singoli oggetti.

potrebbe essere che $\mathcal{I}_C$ corrisponda agli oggetti A e B che hanno lasciato cadere un terzo oggetto E per terra; potrebbe addirittura accadere (come in fig. 4.1) che B consti di due persone, e allo *split* esca una persona della coppia originale(rimescolando dunque gli oggetti). Ci si precluderà questa via nel seguito della presente sezione.

Nota 4.6.1 *Una funzione di matching* ideale *dev'essere in grado di dare più risposte possibili,* anche *risposte negative (di tipo* "don't know"*). Ci sono casi irrisolubili (per cui è meglio non dire niente) e soprattutto casi che non siamo ancora in grado di risolvere ma che tra qualche frame potremo sperare di saper risolvere; in entrambi i casi è bene che la funzione di matching sappia* non *prendere posizione.*

Il modello adottato per risolvere il problema è il seguente: si prende una funzione di scelta tra *una sola* immagine $\mathcal{I}$ e due sequenze $(\mathcal{S}_A, \mathcal{S}_B)$, che dica con quale 'probabilità' l'immagine appartiene alla prima o alla seconda (in modo tale che la somma tra le due sia unitaria:

$$\tilde{\mathbf{f}}(\mathcal{I}, \mathcal{S}_A, \mathcal{S}_B) = [p_A, p_B]; \qquad p_A + p_B = 1 \tag{4.30}$$

Nota 4.6.2 *Il* notevolissimo *vantaggio di questo approccio (che giustifica in buona misura buona parte delle scelte fatte in questa tesi) è che la funzione* $\tilde{\mathbf{f}}$ *potrà discriminare tra l'immagine e le sequenze* entrando *nel merito dei singoli descrittori. In altre parole, dovendo accomunare un'immagine a più sequenze, si*

può osservare quali descrittori **caso per caso** *siano più atti a distinguere le sequenze proposte. Se due sequenze hanno parametri simili (i.e. due persone) e poche differenze (i.e. il contrasto di colore), è possibile sintetizzare con facilità una distanza che volta per volta sfrutti le diversità tra i campioni, passando sopra alle qualità comuni. In qualunque altro caso, una distanza non ideale eppure ben fatta avrebbe rivelato una distanza assoluta pressoché nulla tra le due sequenze dell'esempio, e una metrica 'globale' non avrebbe potuto produrre altro che risultati simili dal matching tra l'immagine e le due sequenze.*

Per risolvere il problema di cui sopra, sarà sufficiente applicare la eq. (4.30) a entrambe le immagini e ottenere così una matrice di scelta:

$$M_{2,2} = \begin{bmatrix} \tilde{\mathbf{f}}(\mathcal{I}_C, \mathcal{S}_A, \mathcal{S}_B) \\ \tilde{\mathbf{f}}(\mathcal{I}_D, \mathcal{S}_A, \mathcal{S}_B) \end{bmatrix} = \begin{bmatrix} p_1 & 1-p_1 \\ p_2 & 1-p_2 \end{bmatrix} \tag{4.31}$$

Vi sono possibili interpretazioni per la matrice: in prima approssimazione potremmo prendere la matrice come *buona* se dà risultati coerenti (cioé se f avvicina le due immagini a sequenze diverse) e come cattiva se dà risultati incoerenti (entrambe le immagini vengono accomunate a una solo sequenza). In ultima analisi si potrebbe pensare di usare soglie ancora più restrittive e di non prendere decisioni nel limbo che si viene a creare in mezzo.

Di questo si parlerà più approfonditamente nel paragrafo successivo.

4.7 Matching tra m Immagini e n sequenze.

Una naturale estensione del modello del paragrafo precedente è di aumentare il numero di possibili immagini e sequenze coinvolte nel *matching*.

Siano date m immagini $\mathcal{I}_1$, ..., $\mathcal{I}_m$ e n sequenze $\mathcal{S}_1, ..., \mathcal{S}_n$. La più generica funzione di matching potrebbe essere del tipo:

$$\mathsf{T}(\mathcal{I}_1, ..., \mathcal{I}_m, \mathcal{S}_1, ..., \mathcal{S}_n) = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{m1} & p_{m2} & \cdots & p_{mn} \end{bmatrix}, \tag{4.32}$$

laddove $\mathsf{T_{ij}}$ sia rappresentativa di un *fattore di matching* [16] (chiamarlo probabilità non sarebbe corretto poiché non si son fatte ancora ipotesi di normalizzazione) tra l'immagine i^{ma} e la sequenza j^{ma}.

Definizione 4.7.1 (Coerenza di una Matrice di Matching) *Una matrice di matching* **quadrata** *T si dice* coerente *se:* [17]

$$\forall i_1, i_2 \left(c_{\bar{i_1},\bar{j_1}} = \max_j(c_{\bar{i_1},j}), c_{\bar{i_2},\bar{j_2}} = \max_j(c_{\bar{i_2},j}), \Longrightarrow j_1 \neq j_2 \right) \tag{4.33}$$

Ancora una volta si è deciso di separare il problema in sotto-parti più facili da gestire, perdendo in espressività ma guadagnando in semplicità d'implementazione e modularità. *Il calcolo effettivo della matrice è stato ridotto (per righe) a m calcoli di matching tra un'immagine singola $\mathcal{I}_i$ e tutte le n sequenze $\mathcal{S}_1, ..., \mathcal{S}_n$.*

Questo ha il notevole vantaggio di poter definire una metrica molto semplice – analogamente alla eq. (4.30) – del tipo:

$$\tilde{\mathbf{f}}(\mathcal{I}, \mathcal{S}_1, ..., \mathcal{S}_n) = [p_1, p_2, \ldots, p_n]; \qquad \sum_{k=1}^{n} p_k = 1 \tag{4.34}$$

che risponde alla domanda: "a quale delle sequenze $\mathcal{S}_1, ..., \mathcal{S}_n$ somiglia di più l'immagine $\mathcal{I}$?" Data questa funzione, la matrice T (eq. (4.32)) è semplicemente ricondotta alla eq. (4.34) tramite:

$$\mathsf{T}(\mathcal{I}_1, ..., \mathcal{I}_m, \mathcal{S}_1, ..., \mathcal{S}_n) = \begin{bmatrix} \tilde{\mathbf{f}}(\mathcal{I}_1, \mathcal{S}_1, ..., \mathcal{S}_n) \\ \tilde{\mathbf{f}}(\mathcal{I}_2, \mathcal{S}_1, ..., \mathcal{S}_n) \\ \ldots \\ \tilde{\mathbf{f}}(\mathcal{I}_m, \mathcal{S}_1, ..., \mathcal{S}_n) \end{bmatrix} \tag{4.35}$$

Lo svantaggio è che sono state ridotte le potenzialità della matrice: non c'è correlazione tra le righe; in altre parole non si fa uno studio comparato tra immagini e sequenze ma solo tra le sequenze. Se si volesse rispondere, per esempio, a un problema del tipo: "Un oggetto si divide in due" la matrice 2×1 che ne deriverebbe sarebbe *sempre* $[1\ 1]^T$, quindi non sarebbe latrice di alcuna informazione. In generale, quando la matrice abbia un numero di righe superiore al numero di colonne, il modello diventa meno espressivo.

[16] Un valore alto indicherà comunque buona probabilità di matching e un fattore nullo probabilità nulla.

[17] ovvero se gli elementi massimi di ogni riga hanno colonne sempre diverse.

Abbiamo però ritenuto che casi del genere siano meno interessanti, poiché un obiettivo del Tracking è proprio di non 'perdere di vista' gli oggetti, quindi l'attenzione è maggiormente rivolta – nel nostro caso – alle sequenze $\mathcal{S}_1, ..., \mathcal{S}_n$. In più, i vantaggi pratici di quest'approccio sono innegabili.

4.7.1 Possibile uso della *Matrice di Matching*

É inevitabile cercare un'interpretazione per la matrice che porti da un insieme di numeri '*fuzzy*' a un certo insieme determinato di *scelte*.

Occorre (nota (4.6.1)) cercare una soluzione già al primo frame, seguendo un compromesso tra i due seguenti vincoli contrastanti:

- É necessario convergere a una soluzione decisa quanto prima, poiché le immagini in output potranno *presto* essere coinvolte in nuovi *merge*.
- Più frame si attendono per la decisione più dati si hanno a disposizione per una corretta classificazione.

Il modello proposto è del tipo: a ogni frame calcolo la matrice e prendo delle decisioni del tipo: $\mathcal{R}(I_j, \mathcal{S}_k)$. Queste possono chiudere interamente il problema o lasciarlo parzialmente irrisolto. In questo caso si attenderanno gli sviluppi successivi consci che il problema da gestire sarà tanto più semplice quante più scelte saranno state fatte.

Definizione 4.7.2 (Interpretazione della Matrice di Matching) *Data una Matrice di Matching T (di dimensioni $M \times N$), si definisce possibile* Interpretazione *una qualunque matrice $I[T]$ ($M \times N$) a valori in $\{0;1\}$ che rispetti i seguenti vincoli:*[18]

$$\forall i \in [1..m] : \qquad \sum_{j=1}^{n} I_{ij} \leq 1 \tag{4.36}$$

Una riga che contenga un uno verrà detta **associata**. *Una riga con soli zeri verrà detta* **non associata**. *Infine, diremo che la sequenza $\mathcal{S}_j$ è* **associata** *all'immagine $\mathcal{I}_i$ (o alla riga i^{ma}) per una certa interpretazione $I[T]$ se $I_{ij} = 1$.*[19]

[18] L'eq. (4.36) esprime semplicemente il fatto che la matrice deve contenere al più un *uno* su ogni riga; tutte le altre celle devono essere a *zero*.

[19] Si noti che ogni immagine può essere associata al più a una sequenza (mentre ogni sequenza può essere associata a più immagini).

Più avanti si confonderà la matrice $I[T]$ con il metodo di estrazione della matrice stessa in funzione di T.

Il significato associato a un'*Interpretazione* è il seguente:

- se la riga i^{ma} è *associata* a una sequenza $\mathcal{S}_j$, abbiamo *deciso* che l'immagine $\mathcal{I}_i$ è la nuova istanza della sequenza $\mathcal{S}_j$;
- se la riga i^{ma} è *non associata*, non si è stabilita alcuna relazione tra l'immagine $\mathcal{I}_i$ e le sequenze. La decisione viene demandata a un tempo successivo.

Si noti che a ogni decisione (associata a I_{ij}) presa, si può semplificare la matrice originale T in un sotto-problema più semplice (prendendone il minore M_{ij}). Questo approccio, insomma, ben si presta a un'iterazione dei calcoli (ovviamente le distanze computate funzioneranno meglio se vengono tolti elementi dal problema).

Prima di procedere con una proposta di *interpretazione*, si osservi il seguente esempio. Siano date le due matrici 3×3 (ometto per brevità gli argomenti):

$$\mathsf{T}_1 = \begin{bmatrix} 0,1 & 0,1 & 0,8 \\ 0,8 & 0,1 & 0,1 \\ 0,1 & 0,8 & 0,1 \end{bmatrix}, \qquad \mathsf{T}_2 = \begin{bmatrix} 0,423 & 0,210 & 0,367 \\ 0,753 & 0,091 & 0,156 \\ 0,230 & 0,208 & 0,562 \end{bmatrix}, \tag{4.37}$$

La matrice T_1 è una caso ideale di matrice *coerente* (eq. (4.33)) che può essere senz'ombra di dubbio letto come: le immagini $\mathcal{I}_1, \mathcal{I}_2, \mathcal{I}_3$ corrispondono alle sequenze $\mathcal{S}_3, \mathcal{S}_1, \mathcal{S}_2$ (rispettivamente).

Il caso T_2 è senz'altro più interessante, in quanto più problematico. Un modo di leggere la matrice data, procedendo con ordine, può essere: $\mathcal{I}_1$ è vista molto vicina a $\mathcal{S}_1$, ma (purtroppo) anche a $\mathcal{S}_3$; $\mathcal{I}_1$ è *molto* legata a $\mathcal{S}_1$ e questo è senz'altro il dato più eclatante; $\mathcal{I}_3$ è molto vicina a $\mathcal{S}_3$. Poiché questa seconda matrice è incoerente (secondo l'eq. (4.33)), si vede chiaramente che si possono avere responsi diversi a seconda di come la si vuole leggere.

In entrambi casi le valutazioni sono qualitative; per passare a euristiche quantitative occorre fissare delle soglie che discriminino tra certezza e incertezza (anche T_1 può essere vista come *incerta*, con un'opportuna soglia!). Abbiamo visto come gestire le certezze e le incertezze.

Un semplice algoritmo che cerchi sempre l'elemento massimo può essere:

Definizione 4.7.3 (Algoritmo di scelta del massimo *senza* ricalcolo) *Data T di dimensioni $(M \times N)$ (con $M \leq N$), costruisco I secondo le seguenti regole:*

Definizione ricorsiva di $f(T, I)$.

Sia T_{ij} il valore massimo di T e θ la soglia di decisione.

if $(T_{ij} < \theta)$ { *poni l'intera matrice I a zeri;*}

else if $(M == 1)$ { *poni I_{1j} a uno e le altre celle a zero;* }

else { *poni I_{ij} a uno e tutti gli altri valori della stessa riga e colonna a zero. Siano $M_{ij}(T)$ e $M_{ij}(I)$ i minori delle matrici T e I (rispettivamente) ottenuti togliendo da dette matrici la riga i^{ma} e la colonna j^{ma}. Riempi gli elementi mancanti della matrice I chiamando ricorsivamente $f(M_{ij}(T), M_{ij}(I))$;* }

Questo semplice algoritmo cerca sempre l'elemento maggiore (ritenuto più certo); se sotto soglia, vuol dire che non vi sono (più) valori che motivino una scelta. Se invece il massimo supera la soglia, viene presa *una* decisione per quell'immagine e quella sequenza. Se l'immagine non era l'ultima, si continua sul resto della matrice dimenticando tutte le righe e colonne già coinvolte in una qualche scelta.

Esistono certamente algoritmi migliori; per esempio, si potrebbe cercare quella permutazione della matrice T [20] con somma massima dei valori (che, come insegna la *Ricerca Operativa*, non necessariamente contiene il massimo assoluto): l'algoritmo proposto sembra comunque sensato poiché la migliore decisione locale (massimo valore singolo di T_{ij}) è pur sempre la meno incerta, e non si può trascurare la semplicità d'implementazione (che consente tra l'altro la ricorsione). Occorre riflettere sul fatto che se fosse davvero necessario scomodare algoritmi di massimizzazione della Ricerca Operativa, la matrice non potrebbe essere coerente; avrebbe dunque senso non prendere decisioni. La variante che segue giustifica ulteriormente la scelta fatta:

Definizione 4.7.4 (Algoritmo di scelta del massimo *con* ricalcolo) *Data T di dimensioni $(M \times N)$ (con $M \leq N$), si costruisce I secondo le seguenti regole:*

Definizione ricorsiva di $f(T, I)$.

[20] qualora non sia diagonale, basta prendere tutte le possibili matrici quadrate con numero massimo di righe tra cui scegliere: l'idea non cambia, solo la formulazione.

Sia T_{ij} il valore massimo di T e θ la soglia di decisione.
if $(T_{ij} < \theta)$ { *poni l'intera matrice I a zeri;*}
else if $(M == 1)$ { *poni I_{1j} a uno e le altre celle a zero;* }
else { *poni I_{ij} a uno e tutti gli altri valori della stessa riga e colonna a zero. Sia T' la matrice ottenuta ricalcolando la* Matrice di Matching *togliendo a T gli argomenti $\mathcal{I}_i$ e $\mathcal{S}_j$* [21] *Sia $M_{ij}(I)$ il minore delle matrice I ottenuto togliendole la riga i^{ma} e la colonna j^{ma}. Riempi gli elementi mancanti della matrice I chiamando ricorsivamente $f(T', M_{ij}(I))$;* }

Come si può notare questa seconda versione differisce solo per il fatto che per ogni *scelta* fatta, ovvero ogni accoppiamento immagine-sequenza, la matrice T viene ricalcolata. Questo ha il notevole *vantaggio* che – essendoci meno variabili in gioco – la distanza comparata può far maggiore differenza sulle features dei 'pochi rimasti'; esperimenti di questo tipo hanno confermato la bontà dell'algoritmo. Il rovescio della medaglia è che un ricalcolo a ogni iterazione è meno *ingegneristico* da un punto di vista del *software*: non si trascuri che poter associare una ed una sola matrice ad un *fenomeno di tracking* (che possa essere trasportata tra i vari moduli del programma) può a volte essere un pregio irrinunciabile, soprattutto se il decisore è a un livello più alto e fa scelte che a questo livello non possono essere fatte. In altre parole, l'algoritmo di decisione ne guadagna, ma a volte potrebbe essere necessario *esportare* la sola matrice originale T.

Per quel che concerne la soglia θ,sono state fatte prove sperimentali *di decisione* su matrici 2×2 e 3×3. Per essi una buona soglia si è rivelata essere:

$$\theta_2 = 0.6; \qquad \theta_3 = 0.38; \tag{4.38}$$

In generale, data una matrice T $(M \times N)$, si può pensare di fare dipendere la soglia solo da N[22], in virtù del principio di separazione della matrice per righe (eq. (4.35)) descritto e in parte giustificato alla fine della Sez. 4.7 (pag. 63). Poiché in caso di parità (caso peggiore) ogni coefficiente vale $1/N$, la soglia dovrà essere presa superiore a questo numero; **più alta la soglia più prudente è l'algoritmo di scelta** (ovvero vengono fatte meno scelte, in favore di una soluzione

[21] ovvero applicando la definizione al sottoproblema ottenuto togliendo l'immagine e la sequenza appena messi in relazione – e quindi usciti dal problema.

[22] ovvero la *larghezza* della matrice.

in un tempo successivo). Un buon compromesso può essere:

$$\theta_N = \frac{1,2}{N} \tag{4.39}$$

Si potrebbe studiare un sistema a *soglia adattativa* che riconosca se l'ambiente è più o meno rumoroso e decida la soglia di conseguenza a decisioni precedenti: le collisioni possono essere poco frequenti, consentendo un *late binding*; mentre in altri ambienti si può essere costretti a scegliere in fretta poiché i dati cambiano spesso. Il discorso ha perfettamente senso poiché prima di entrare in conflitto le sequenze vengono tipicamente inseguite senza alcun problema per decine di frame; in ciascuno di questi momenti si può fare un *matching virtuale* per tarare il sistema (in funzione del rumore o anche solo della somiglianza degli oggetti in fase di analisi). Questo è un possibile sviluppo futuro.

É stata infine studiata una metrica per valutare la **bontà** di una matrice di matching, utile in fase di sperimentazione (vedere Cap. 5). Essa è definita come il valore massimo assunto da una permutazione della matrice normalizzato per la somma di tutti questi valori. Segue una definizione utile al calcolo della metrica per matrici *larghe*, ovvero con $N \geq M$.

Definizione 4.7.5 (Permutazione di una matrice 'larga') *Data una matrice $\mathcal{A}$ di dimensione $M \times N$ ($N \geq M$) di componenti $a_{i,j}$ si definisce permutazione di $\mathcal{A}$ una lista $l \equiv (a_{1,p(1)}, a_{2,p(2)}..., a_{M,p(M)})$, ove $p : \mathbb{N}_M \longmapsto \mathbb{N}_M$ è un applicazione* biiettiva. *In particolare definiremo* **valore** *di una permutazione il prodotto delle celle:*

$$v(p) \doteq \prod_{i=1}^{M} a_{i,p(i)} \tag{4.40}$$

Ora, la metrica adottata può dunque essere agevolmente definita come:

$$\beta(\mathcal{A}) \doteq \frac{\max\limits_{p_i(\mathcal{A}) \in \mathcal{P}(\mathcal{A})} |v(p_i)|}{\sum\limits_{p_i(\mathcal{A}) \in \mathcal{P}(\mathcal{A})} |v(p_i)|} \tag{4.41}$$

dove $\mathcal{P}(\mathcal{A})$ è lo spazio (finito) di tutte le possibili permutazioni di $\mathcal{A}$ e $\beta(\mathcal{A})$ è appunto la **bontà** della matrice stessa. Essa assume valori tra $\frac{1}{C}$ e 1 (dove C è la cardinalità di $\mathcal{P}(\mathcal{A})$). In particolare:

- assume valore minimo per una matrice con tutti i valori uguali; per esempio in:

$$\begin{bmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix}$$

  si avrebbe $\beta = \frac{1}{6}$ (il minimo);

- assume valore massimo (uno) per una matrice con tutti i valori della permutazione diversi da zero e tutte le altre celle a zero, come per esempio:

$$\begin{bmatrix} 0 & 0 & 7 \\ 0 & 6 & 0 \end{bmatrix}.$$

  Si noti dall'esempio che la definizione funziona *anche* per matrici i cui valori su ogni riga non siano normalizzati.

Nei due casi presi in esame a inizio sezione, i valori sarebbero:

$$\beta(\mathsf{T}_1) = 0.9517 \qquad \beta(\mathsf{T}_2) = 0.4513,$$

ovvero due risultati perfettamente in accordo con lo studio che se n'è fatto.

Questa definizione tornerà maggiormente utile nel Cap. 5.

4.8 Matching tra m sequenze e n sequenze (cenni).

Una naturale estensione cui si potrebbe pensare è di vedere crescere le immagini (al primo frame successivo allo *split*) in vere e proprie sequenze, qualora – per esempio – non fossimo stati in grado di classificare con successo gli oggetti uscenti dallo *split* al primo tentativo. L'informazione a disposizione per gli oggetti "del dopo" cambia e a poco a poco cresce in una vera e propria *statistica*.

Per quanto la definizione di un nuovo modello possa essere semplice, non si è seguita questa strada da un punto di vista implementativo; si è invece optato per iterare più volte l'eq. (4.32) in K frame successivi e applicare semplici considerazioni sulle K matrici derivanti per trarre delle decisioni.

É certamente uno spunto interessante per sviluppi futuri.

4.9 Conclusioni

Sono state presentate in questo capitolo molte possibili metriche tra *immagini* e *sequenze*. Mano a mano che si cresce con la complessità (immagine singola $\hookrightarrow$ insieme di immagini), vi sono più dati a disposizione per la sintesi si una funzione di matching.

Sono state presentate metriche 'scolastiche' (metrica in $||\ \cdot\ ||_1$ e in $||\ \cdot\ ||_2$) – eq. (4.9) e eq. (4.10), con $K = 1, K = 2$ rispettivamente – che si sono dimostrate sperimentalmente piuttosto limitate, per la discriminazione troppo alta tra oggetti simili.

Sono state proposte metriche alternative a soglia (a $K\,\sigma$) – Sez. 4.4.2, Sez. 4.4.3 – che si sono mostrate buone sia nella diversità che nella somiglianza tra sequenze.

Sono stati introdotti ulteriori strumenti che si poggiassero su distanze elementari al fine di mettere in relazione tra loro più parti coinvolte in una *fusione* che ne abbia reso impossibile il tracking per un certo numero di frames. É stata dunque definita un'importante Matrice di Matching che riunisce in sè tutte le possibili distanze tra oggetti coinvolti.

É stata infine sviluppata una proposta d'uso per la suddetta matrice, in modo da poter integrare in un generico modulo di Tracking un apparato di scelta in casi di *merge/split*.

Capitolo 5

Risultati Sperimentali

God does not play dice with the universe.

A. Einstein

5.1 Introduzione

In questa sezione verranno introdotti gli strumenti usati per gli esperimenti e soprattutto le assunzioni fatte (con relativa giustificazione).

Sono state fatte numerose prove su due data-set:

- sequenze prese da **PETS2001** (*Performance Evaluation of Tracking and Surveillance* [1]); nel data-set sono coinvolte persone e veicoli;
- sequenze registrate nel parco della facoltà (già usate per testare il sistema di *foreground-detection*), in cui vengono coinvolte solo persone.

Entrambi i data-sets sono stati progettati per fornire problematiche diverse (cambiamenti di illuminazione, incroci e somiglianze tra oggetti).

[1] Si tratta di workshop che hanno l'obiettivo di promuovere la valutazione delle prestazioni ed il **confronto** degli algoritmi di visione impiegati nei sistemi di videosorveglianza, con particolare riferimento agli algoritmi di tracking che tipicamente di tali sistemi costituiscono il nucleo fondamentale. A tale scopo gli organizzatori mettono a disposizione dei potenziali autori alcuni data-set (che sono stati appunto usati nel lavoro svolto) che devono obbligatoriamente essere utilizzati per presentare i risultati sperimentali degli algoritmi proposti. In particolare, PETS2001 è stato il secondo Workshop promosso; questi stanno avendo cadenza annuale. Vedere [3].

Si ricorda che, nonostante il lavoro sia stato fatto off-line, i dati di partenza non sono stati segmentati "a mano" ma è stata usata la segmentazione (*imperfetta*) del Sentinel.

La difficoltà del sistema sta nel riconoscere correttamente figure umane. Le basi stesse del lavoro svolto nascono dall'ipotesi di discriminare figure umane (tramite *appearance models*) in un sistema che era in grado di gestire solo forme e bordi, quindi assolutamente 'cieco' nei confronti di aspetti quali tessitura, colori, eccetera.

Sono state usate sequenze di lunghezza variabile ($10 \div 400$ immagini, spesse volte sono state spezzate in blocchi da 100). Per avere un proliferare di dati per ogni esperimento, è stata adottata la seguente filosofia:

- per lo studio di un'*immagine* $\mathcal{I}$ si sono prese istanze successive di una sequenza ($S_\tau(0)$, $S_\tau(1)$, ...) e se n'è osservata la stabilità nel tempo;
- per lo studio di una *sequenza* $\mathcal{S}_\tau$ si son prese tutte le sequenze 'parziali' ($\mathcal{S}_0$, $\mathcal{S}_1$, ...) simulando una crescita frame per frame della sequenza stessa. In questo caso si può osservare una stabilizzazione nel tempo verso un giudizio esatto, il tempo di convergenza a conseguirlo, la stabilità a regime della risposta.

Offrendo non uno ma cento risultati per esperimento, questo metodo ha inoltre il vantaggio di valutare la bontà di un modello anche da un punto di vista quantitativo (*"dice il vero il tot% delle volte"* piuttosto che *"ci ha preso o meno"*).

Ciò si è reso necessario poiché gli studi fatti offrono poche possibilità di introdurre *metriche di performance* per misurare e confrontare la bontà degli esperimenti. Sono stati volta per volta introdotti coefficienti che, uniti alla conoscenza umana del problema, possono garantire un'effettiva quantificazione della *resa* degli strumenti usati. Se ne parlerà sezione per sezione.

Si noti peraltro che tutti questi discorsi hanno senso solo da un lato pratico: è più comodo vedere tanti numeri (tanti piccoli esperimenti uguali su campioni simili) che uno solo; ma si ricordi che l'esperimento più significativo è *comunque* quello con la *prima* immagine di una sequenza, ovvero il primo. L'interesse decresce col tempo poiché dopo non necessariamente ha senso confrontare una sequenza con un immagine 'vecchia' di 100 frame (l'oggetto potrebbe allontanarsi,

girarsi e quindi perdere la costanza di molte features, che invece appena dopo un merge dovrebbero essere più vicine all'originale).[2]

Nota. Si farà uso della notazione $\boxed{ABC \Rightarrow DE}$ per indicare che gli oggetti A, B e C sono coinvolti nel merge e che dallo split escono D ed E. Si userà la notazione A' per indicare una sequenza che è simile ad A (ovvero una coppia di oggetti che un umano riconoscerebbe come 'uguali'). Si noti che vi può essere un'enorme differenza tra i due oggetti come forma, dimensioni, posizione, a causa di un possibile contesto temporale diverso.

Si parlerà in generale di sequenze/immagini di **input** (o *before-images*) per intendere gli oggetti coinvolti nel merge, che quindi entrano nel sistema. Si parlerà invece di sequenze/immagini di **output**[3] (o *after-images*) per intendere gli oggetti che escono dallo split. L'asimmetria del sistema nasce dal fatto che i primi hanno – al momento dello studio – una storia completa e statica, i secondi una storia inizialmente nulla e in crescita.

5.2 Match tra N sequenze d'ingresso e M immagini in uscita

Questa classe di esperimenti sfrutta la *Matrice di Matching* introdotta a pag. 63. Il problema da studiare è: N corpi vengono correttamente inseguiti (*tracciati*) fino all'istante t_0. In questo momento, avviene la fusione (*merge*) tra questi e in un momento t_1 (in attesa del quale non viene svolto alcun calcolo) alcuni oggetti si dissociano (*split*). Non sappiamo la relazione tra i *before-objects* e gli *after-objects*: nella migliore delle ipotesi $N = M$ e basterà trovare chi corrisponde a chi, ma in casi reali occorre ammettere la possibilità di nascite, morti e addirittura rimescolamenti (vedere (fig. (4.1), pag. 62)).

Per le matrici sono stati ideati dei parametri di rendimento che tentano di valutare la buona riuscita dell'esperimento.

[2]Purtroppo, il tempo di fusione potrebbe essere così alto da invalidare *anche* questo tipo di discorso.

[3]Nel senso informatico del termine, tuttavia, sono entrambi input!

Sia T il numero totale di frame calcolati.[4] In ogni esperimento viene calcolata una matrice per tipo (stazionario o meno) e per frame, per un totale di $2T$ matrici.

I parametri che seguono dipendono dalla stazionarietà e sono espressi per ogni frame:

Coerenza lasca, δ_{CL} Attributo booleano di una matrice che vale se è coerente ((Def. (4.7.1), pag. 64)), ovvero se i massimi di ogni riga stanno su colonne diverse. Quando una matrice è coerente, *esiste* una permutazione di M che definisce una scelta 'ovvia' per correlare ingressi e uscite. In altre parole, essa ha una sola scelta ottima rispetto a qualunque criterio.

Coerenza stretta, δ_{CS} Attributo booleano applicabile a una matrice coerente (in senso lasco); è vero se ha tutti i valori della 'migliore' permutazione sopra soglia (verrà usata la soglia di (eq. (4.38), pag. 68)).[5]

Bontà, β Il valore assunto da β (vedere (eq. (4.7.1), pag. 69)). La bontà esprime quanto la scelta migliore primeggi rispetto alle altre per la matrice in questione.

Sull'esperimento nella sua globalità, verranno usati:

• $\overline{C_l\%}$: numero di volte (espresso in percentuale) in cui la matrice è coerente in senso lasco.

• $\overline{C_s\%}$: numero di volte (espresso in percentuale) in cui la matrice è coerente in senso stretto. Si noti che $\overline{C_s\%} \leq \overline{C_l\%}$.

• **Bontà media,** $\overline{\beta}$: il valore medio assunto da β nei T frame.

• **Matrice media,** $\overline{M}$: si prenderà in esame la matrice che ha su ogni cella il valore medio assunto nei T frame.

• **Risultato atteso,** $\overline{R_a}\%$: numero di volte che la matrice esprime esattamente tutte e sole le scelte attese.

• **Altro (**$\overline{R_a^*}\%, ...$**)**: verranno volta per volta definiti parametri ritenuti utili per l'esperimento in questione, tipicamente indicativi della sua corretta riuscita.

[4] il numero di frame è, per motivi di praticità, pari alla lunghezza minima delle sequenze di output. Per avere grafici della stessa lunghezza sono stati tagliati quelli più lunghi, ovviamente perdendo le sequenze più *lontane* dallo split.

[5] la definizione è ben posta poiché per una matrice C.L. la scelta ottima *è* quella dei valori massimi di ogni riga (poiché sono su colonne distinte).

In particolare, $\overline{R_a^*}\%$ ha senso solo per matrici in cui la soluzione comporti una scelta completa (tutte le after-images vengono correlate a una sequenza d'input). E' un indicatore del tutto simile a $\overline{R_a}\%$ con la differenza che viene rilassato il vincolo della soglia. Si può leggere come: numero di volte in cui la migliore permutazione è quella giusta.

Si noti che solo gli ultimi due parametri esprimono una conoscenza a posteriori sull'esperimento; per gli altri, una matrice può avere valori buoni ma su associazioni sbagliate; questo vuol dire che la matrice è molto 'convinta' delle scelte fatte, non che siano giuste. Questa spia indicherà o un errore di modello o un esperimento particolarmente critico (quindi un *limite* del modello).

L'algoritmo di scelta che verrà sempre usato è – per semplicità di notazione – l'algoritmo senza ricalcolo (Def. (4.7.3), pag. 67).[6]

Poiché in casi reali queste matrici sono piccole, sono state fatte prove con poche possibili combinazioni. Seguono esperimenti con matrici di tipo: 3×3. Più avanti vi sono altri esperimenti con matrici 2×2 (pag. 80) e 2×3 (pag. 86).

5.2.1 Esperimenti con matrici 3×3

Queste matrici sono certamente più problematiche poiché col crescere della dimensione diminuisce la distanza tra il primo e il secondo di ogni riga, rendendo dunque difficile distinguere correttamente le associazioni e vedendole molto meno robuste al rumore. Di solito è difficile attendersi un buon risultato in assoluto dalla matrice (una permutazione con tutti i valori sopra soglia) poiché le scelte passano da 2 (del caso 2x2) a 6; spesso dunque il sistema 'globalmente' fallirà ma si potranno comunque dedurre informazioni spesso corrette ammesso che si facciano delle ipotesi supplementari. Se per esempio si impone che vengano fatte sempre tre scelte, il discorso soglia crolla ed è sufficiente uno studio comparato tra le sei diagonali.

La soglia usata è 38 ($\theta = 0,38$, con normalizzazione a 100).

[6] per quanto ci sembri meno potente, sarebbe dispersivo mostrare per ogni scelta fatta le sottomatrici che si ottengono...

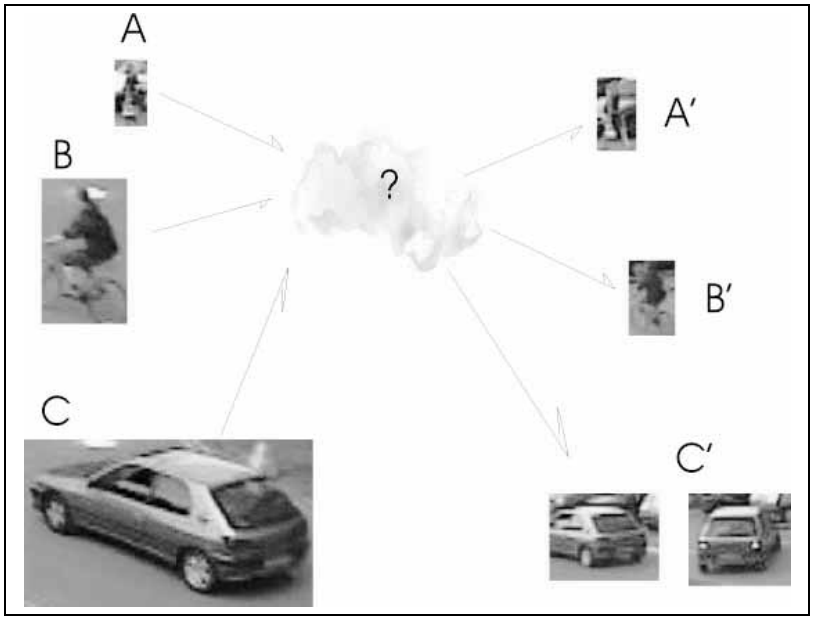

Figura 5.1: Schema del merge-split dell'esperimento 3x3.A.

- **Esperimento 3x3.A**

In questo esperimento sono stati incrociati in modo virtuale tre oggetti: una persona, una bicicletta e un'automobile. L'esperimento è di tipo $\boxed{ABC \Rightarrow A'B'C'}$ (vedere fig. 5.1). Questo caso è molto interessante poiché offre diverse problematiche:

◦ la bicicletta cambia nel tempo orientazione e dimensione;

◦ l'automobile C è la stessa del caso C' ma le due istanze sono riprese a distanze diverse dalla telecamera, con una diversa orientazione e quindi forma diversa. Tutto ciò, unito al fatto che i corpi rigidi sono molto stabili (un corpo più stabile vede distanza più grandi, con gli algoritmi di matching scelti), in parte può giustificare il fatto che il sistema, come si vedrà, fatica a riconoscere C' in C.

Si noterà che, nonostante la matrice sia problematica, si può estrarre comunque dell'utile informazione.

$$\mathcal{M}_S(0) = \begin{bmatrix} \mathbf{61,39} & 12,21 & 26,40 \\ 16,32 & \mathbf{60,58} & 23,10 \\ 28,40 & \mathbf{57,92} & 13,68 \end{bmatrix} \qquad \mathcal{M}_{NS}(0) = \begin{bmatrix} \mathbf{78,66} & 14,63 & 6,70 \\ 12,26 & \mathbf{76,82} & 10,92 \\ 36,37 & \mathbf{58,39} & 5,24 \end{bmatrix}$$

$$\overline{\mathcal{M}}_S = \begin{bmatrix} \mathbf{60,41} & 13,73 & 25,86 \\ 17,61 & \mathbf{63,54} & 18,84 \\ 30,21 & \mathbf{36,64} & 33,15 \end{bmatrix} \qquad \overline{\mathcal{M}}_{NS} = \begin{bmatrix} \mathbf{71,65} & 13,13 & 15,22 \\ 16,85 & \mathbf{73,78} & 9,37 \\ \mathbf{42,69} & 31,00 & 26,31 \end{bmatrix}$$

Applicando l'algoritmo di decisione al tempo zero, il sistema cercherebbe il massimo della matrice, che per entrambi i casi è $\mathcal{M}_{11}$. Farebbe dunque la (corretta) associazione $\Re(A, A')$ poi cercherebbe il massimo del minore $\mathcal{M}_1^1$ ([7]) che è per entrambi – ragionando sulla matrice iniziale – $\mathcal{M}_{22}$. Il sistema farebbe dunque la seconda (corretta) associazione $\mathcal{R}(B, B')$ e osserverebbe che il minore rimasto, collassato a cella singola (la $\mathcal{M}_{33}$) non supera la soglia (il maggiore dei due vale infatti 13,68). Le **interpretazioni** *effettiva* e *teorica* sarebbero dunque (secondo la (Def. (4.7.2), pag. 65)) rispettivamente:

$$T_{eff} = \begin{bmatrix} \mathbf{1} & 0 & 0 \\ 0 & \mathbf{1} & 0 \\ 0 & 0 & \mathbf{0} \end{bmatrix} \qquad T_{teor} = \begin{bmatrix} \mathbf{1} & 0 & 0 \\ 0 & \mathbf{1} & 0 \\ 0 & 0 & \mathbf{1} \end{bmatrix}$$

Riassumendo, il sistema farebbe due associazioni, entrambe corrette, ma non 'vedrebbe' l'ultima. Poiché è stato nelle scelte del nostro gruppo di lavoro avere un sistema che preferisce lasciare irrisolti casi incerti piuttosto che azzardare dei collegamenti, questa risposta è positiva: già scioglie notevolmente il problema (da tante possibilità siamo passati a una sola: C è C' oppure il primo è morto ed è nato il secondo?). Questo nuovo problema può essere lasciato irrisolto (e tentare eventualmente di sviscerarlo nei prossimi frame) o si può tentare un metodo a match singolo per vedere quanto gli oggetti residui siano simili.

Matrici 3x3: esperimento A (61 frame)						
stazionarietà	$\overline{C_l\%}$	$\overline{C_s\%}$	$\beta(0)$	$\overline{\beta}(\%)$	R_a	R_a^* %
✓	26,23	18,03	43,61	50,78	18,03	93,44
–	9,84	6,56	59,2	60,02	6,56	90,16

Dalla tabella si evince che il sistema non riesce correttamente a risolvere il problema *nella sua interezza*. Raramente infatti la terza riga 'sceglie' l'accostamento col terzo oggetto e ancor meno questo valore è sopra la soglia data (in effetti la

[7] ovvero la matrice 2x2 ottenuta togliendo a $\mathcal{M}$ la prima riga e la prima colonna.

terza riga ha l'andamento di una variabile aleatoria, quasi fosse scorrelata con le tre sequenze).

Quindi raramente il sistema sceglie *sua sponte* le tre associazioni giuste (R_a), ma se lo si obbliga a fare una scelta completa, tra le sei possibili, viene quasi sempre (R_a^*) preferita quella giusta.

Le β ci danno un'altra informazione notevole: poiché le diagonali possibili sono sei (quindi mediamente avranno un 16% ciascuna di probabilità) e l'esperimento mostra che quella corretta ha mediamente il 50-60%, c'è sempre un grosso dislivello con le alternative possibili (ovvero la prima ha un alto grado di confidenza).

Ricapitolando: in questo esperimento il sistema *non* fa tutte le scelte giuste, ma sono giuste quelle che fa e se forzato a una scelta completa, questa è quella giusta.

Questo risultato è da considerarsi soddisfacente. Infatti il problema è stato impostato in modo del tutto generale, ammettendo quindi che durante il merge possano verificarsi nascite-morti di oggetti. In tale contesto il sistema può solo risolvere le associazione che giudica certe. Esistono tuttavia applicazioni in cui la conoscenza a priori sullo scenario di riferimento consente di escludere a priori nascite e morti in determinate zone dell'immagine: in questo contesto il sistema può forzare tutte le associazioni e, nell'esperimeno mostrato, il risultato sarebbe stato corretto.

5.2.2 Esperimenti con matrici 2×2

Questi esperimenti rappresentano il caso più frequente, semplice e forse più interessante, in cui due oggetti s'incrociano e occorre stabilire le corrette associazioni quando questi si separano. Negli esperimenti A e B vi sono effettivamente due oggetti che mantengono la loro identità e si osserverà come e se il sistema riesca a tracciarli. Nel caso C invece vi sarà un merge-split virtuale in cui il sistema dovrà fare un'associazione soltanto. I casi D ed E sono analoghi ai casi B e C.

La valutazione quantitativa di un esperimento è diversa a seconda che tutte le scelte vadano fatte o solo alcune. Infatti:

- se ogni riga va associata a una colonna (questo vale per qualunque *matrice*

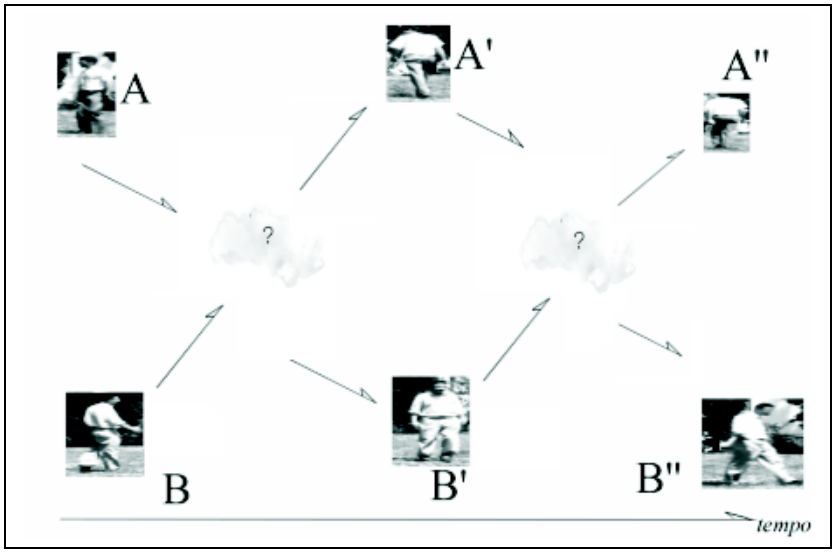

Figura 5.2: Schema degli esperimenti 2x2.A e 2x2.B. Le proporzioni tra le figure sono state conservate.

larga[8], ha senso parlare di coerenza ($\overline{C_l/s\%}$) e bontà (β) della matrice stessa. Questi parametri sono indicativi della supremazia di una scelta sulle altre;

- se invece la riuscita dell'esperimento consiste nel lasciare svincolate alcune righe cambia tutto: si può pensare a una parziale riuscita se vengono fatte scelte che includono tutte le giuste, o a definire una soglia per cui le scelte sono giuste in una certa percentuale (ammesso che esista una soglia abbastanza alta per tagliare le decisioni sbagliate ma tenere le giuste). Fondamentalmente l'unico studio sensato è quello di vedere semplicemente quante volte il sistema dice il vero in senso assoluto.

- **Esperimento 2x2.A**

Abbiamo qui un merge/split di tipo: $\boxed{AB \Rightarrow A'B'}$. Questo è **molto** interessante poiché gli oggetti A e B sono persone con vestiti dello stesso colore e di grandezza simile (vedere fig. 5.2); è dunque un caso molto critico di distinzione tra oggetti. Questo è un caso *reale* di merge e split.

[8]ovvero con un numero di colonne non inferiore al numero di righe

Le matrici in questione al tempo 1 sono:

$$\mathcal{M}_S(0) = \begin{bmatrix} \mathbf{74,03} & 25,97 \\ 39,46 & \mathbf{60,54} \end{bmatrix} \qquad \mathcal{M}_{NS}(0) = \begin{bmatrix} \mathbf{75,01} & 24,99 \\ 40,70 & \mathbf{59,30} \end{bmatrix}$$

Mentre le matrici medie valgono:

$$\overline{\mathcal{M}}_S = \begin{bmatrix} \mathbf{53,04} & 46,96 \\ 32,71 & \mathbf{67,29} \end{bmatrix} \qquad \overline{\mathcal{M}}_{NS} = \begin{bmatrix} \mathbf{53,51} & 46,49 \\ 35,46 & \mathbf{64,54} \end{bmatrix}$$

- **Esperimento 2x2.B**

Abbiamo qui un merge/split di tipo: $\boxed{A'B' \Rightarrow A''B''}$ (vedere fig. 5.2), dove in input abbiamo proprio l'output dell'esperimento A. Anche questo è un caso *reale* con l'ulteriore problema che le sequenze in output sono molto corte (5 e 6 frame).

Questa volta le 4 matrici valgono:

$$\mathcal{M}_S(0) = \begin{bmatrix} \mathbf{74,03} & 25,97 \\ 39,46 & \mathbf{60,54} \end{bmatrix} \qquad \mathcal{M}_{NS}(0) = \begin{bmatrix} \mathbf{75,01} & 24,99 \\ 40,70 & \mathbf{59,30} \end{bmatrix}$$

$$\overline{\mathcal{M}}_S = \begin{bmatrix} 48,73 & \mathbf{51,27} \\ 39,34 & \mathbf{60,66} \end{bmatrix} \qquad \overline{\mathcal{M}}_{NS} = \begin{bmatrix} \mathbf{55,72} & 44,28 \\ 42,60 & \mathbf{57,40} \end{bmatrix}$$

Vi è un'ulteriore colonna (R_A^*) con la percentuali di risposte corrette in caso di forzatura di una risposta (viene bypassato il concetto di soglia, semplicemente viene scelta la diagonale migliore).

Il risultato atteso R_a, invece, è basso per entrambi gli esperimenti e c'è da aspettarselo: il modello è tirato al massimo delle sue possibilità. Il sistema *non* riesce a garantire sempre che A corrisponda ad A' e B a B' ma se si decide di forzare una scelta (obbligare il sistema a decidere uno dei due possibili accoppiamenti) questo raggiunge ottimi risultati:

Matrici 2x2: esperimenti A / B (23 / 5 frame)							
esp.	staz.	$\overline{C_l}\%$	$\overline{C_s}\%$	$\beta(0)$	$\overline{\beta}(\%)$	R_a	$R_a^*\%$
A	✓	47,83	47,83	81,39	69,91	47,83	73,91
A	–	47,83	39,13	81,38	67,69	39,13	52,17
B	✓	40	0	59,38	59,45	0	100
B	–	80	0	63,49	62,90	0	100

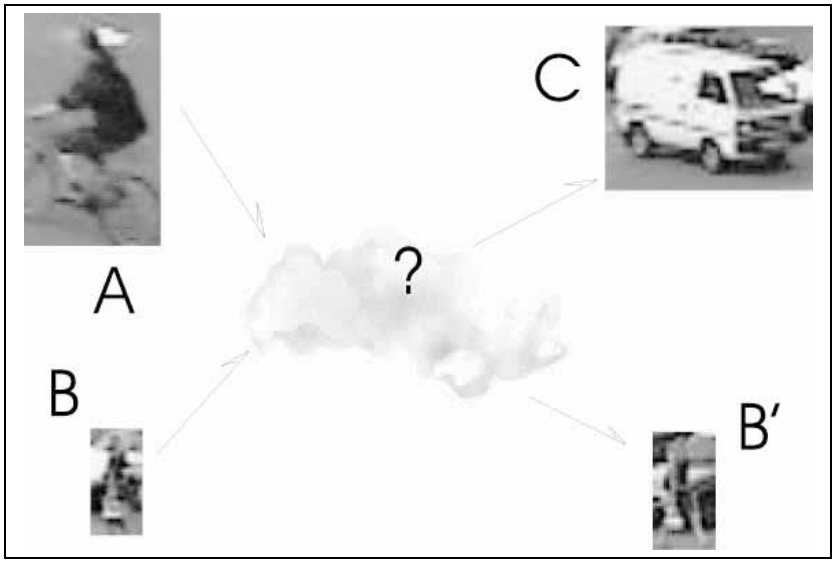

Figura 5.3: Schema dell'esperimento 2x2.C. Le proporzioni tra le figure sono state conservate.

In particolare, nell'esperimento A la risposta è corretta per tutti i primi 12 frame e diventa sbagliata dal 13^{mo} in poi. Questo in pratica significa che (in *questo* caso) un algoritmo prudente farebbe forse una scelta sbagliata, mentre uno più rischioso farebbe la scelta giusta.

- **Esperimento 2x2.C**

Questo esperimento è un esempio di match parziale riuscito: il merge/split è di tipo: $\boxed{AB \Rightarrow CB'}$, (A è una bicicletta, B una donna, C un furgone, vedere fig. 5.3). É un caso virtuale di scontro ma è comunque interessante poiché avvicina due cose uguali a due abbastanza diverse. Si tratta di vedere se il sistema riesce a riconoscerle abbastanza diverse.

Le 4 matrici valgono:

$$\mathcal{M}_S(0) = \begin{bmatrix} 36,57 & \mathbf{63,43} \\ 18,27 & \mathbf{81,73} \end{bmatrix} \qquad \mathcal{M}_{NS}(0) = \begin{bmatrix} 39,97 & \mathbf{60,03} \\ 15,55 & \mathbf{84,45} \end{bmatrix}$$

$$\overline{\mathcal{M}}_S = \begin{bmatrix} 45,89 & \mathbf{54,11} \\ 18,25 & \mathbf{81,75} \end{bmatrix} \qquad \overline{\mathcal{M}}_{NS} = \begin{bmatrix} 49,29 & \mathbf{50,71} \\ 14,07 & \mathbf{85,93} \end{bmatrix}$$

I risultati sperimentali sono:

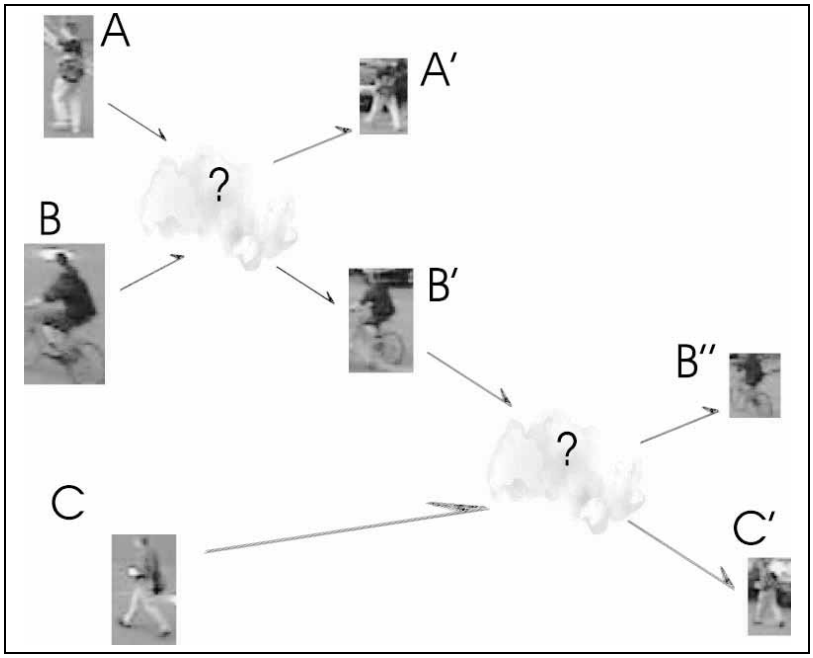

Figura 5.4: Schema degli esperimenti 2x2.D e 2x2.E. Le proporzioni tra le figure sono state conservate.

Matrici 2x2: esperimento C (100 frame)					
stazionarietà	$\overline{C_l\%}$	$\overline{C_s\%}$	$\beta(0)$	$\overline{\beta}(\%)$	R_a %
✓	0,00	0,00	72,07	79,17	95,00
–	74,00	0,00	78,33	85,59	99,00

Buffamente, il sistema risponde bene in modo assoluto (fa la scelta giusta e non sceglie nella seconda possibile associazione) solo dopo il frame iniziale (nel caso stazionario dal sesto in poi, nell'altro addirittura dal secondo); si ritiene dunque di poter aspettare *sempre* qualche frame per fare scelte anche quando queste siano 'forti' (oppure tentare una soglia superiore, ma questo ha certamente degli svantaggi).

In questo caso R_a^* sarebbe stata poco significativa. Non avrebbe avuto senso definirla alla vecchia maniera, piuttosto come: "numero di volte in cui viene scelta l'associazione A_{22} rispetto alla A_{21}"; in tal caso sarebbe stata sempre al 100%.

- **Esperimenti 2x2.D-E**

I due esperimenti che seguono sono tratti da un caso reale di incontro a due a due tra tre oggetti (vedere fig. 5.4). L'esperimento D è del tipo $\boxed{AB \Rightarrow A'B'}$ [9], mentre l'esperimento E è del tipo: $\boxed{B'C \Rightarrow B''C'}$ (che è fondamentalmente la stessa cosa; la notazione è cambiata per rendere l'idea della relazione temporale).

Le matrici valgono:

$$\mathcal{M}_S^D(0) = \begin{bmatrix} \mathbf{74,90} & 25,10 \\ 38,08 & \mathbf{61,92} \end{bmatrix} \qquad \mathcal{M}_{NS}^D(0) = \begin{bmatrix} \mathbf{87,37} & 12,63 \\ 11,78 & \mathbf{88,22} \end{bmatrix}$$

$$\overline{\mathcal{M}}_S^D = \begin{bmatrix} \mathbf{74,88} & 25,12 \\ 32,37 & \mathbf{67,63} \end{bmatrix} \qquad \overline{\mathcal{M}}_{NS}^D = \begin{bmatrix} \mathbf{85,98} & 14,02 \\ 14,60 & \mathbf{85,40} \end{bmatrix}$$

$$\mathcal{M}_S^E(0) = \begin{bmatrix} \mathbf{74,90} & 25,10 \\ 38,08 & \mathbf{61,92} \end{bmatrix} \qquad \mathcal{M}_{NS}^E(0) = \begin{bmatrix} \mathbf{87,37} & 12,63 \\ 11,78 & \mathbf{88,22} \end{bmatrix}$$

$$\overline{\mathcal{M}}_S^E = \begin{bmatrix} \mathbf{75,20} & 24,80 \\ 9,94 & \mathbf{90,06} \end{bmatrix} \qquad \overline{\mathcal{M}}_{NS}^E = \begin{bmatrix} \mathbf{75,95} & 24,05 \\ 9,83 & \mathbf{90,17} \end{bmatrix}$$

Nel complesso si ha:

Matrici 2x2: esperimenti D/E (17 / 63 frame)							
esp.	staz.	$\overline{C_l\%}$	$\overline{C_s\%}$	$\beta(0)$	$\overline{\beta}(\%)$	R_a	$R_a^*\%$
D	✓	100	93,33	82,92	86,16	100	100
D	–	100	100	98,11	97,29	100	100
E	✓	68,85	29,51	96,49	87,95	29,51	98,36
E	–	100	88,52	96,67	93,58	88,52	100

Il caso D viene risolto alla perfezione dal sistema, che invece incontra più difficoltà nel caso E. Le ragione del fatto che il secondo esperimento (apparentemente

[9] non stupisca che l'associazione sia sempre $\boxed{AB \Rightarrow A'B'}$ e mai $\boxed{AB \Rightarrow B'A'}$: sapendo a priori l'accostamento si sono costruite sequenze di questo tipo per comodità di elaborazione tramite fogli Excel.

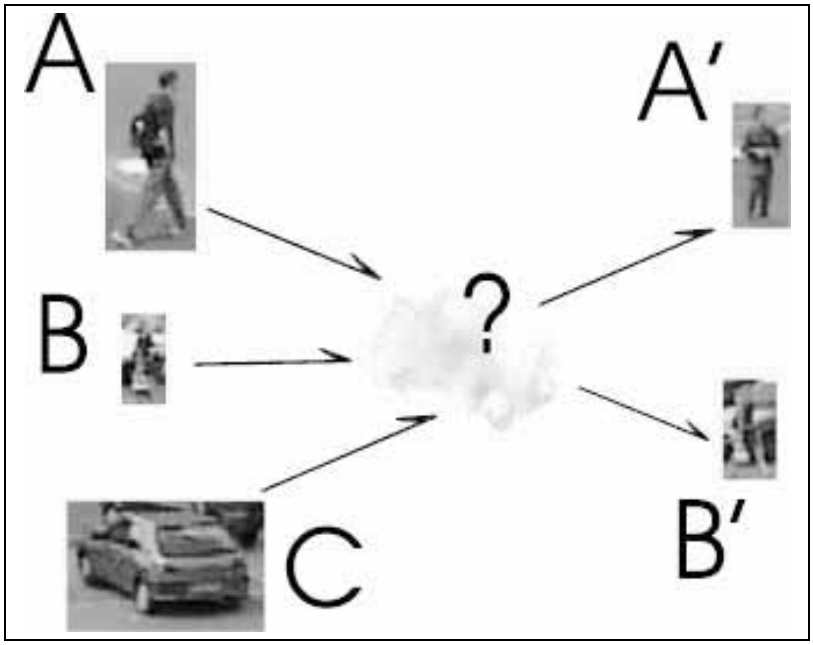

Figura 5.5: Schema dell'esperimento 2x3.A. Le proporzioni tra le figure sono state conservate.

identico al primo) fallisca più spesso sono da ricercarsi nel fatto che nell'ultima sequenza la bicicletta si allontanata ed è più storta per cui comincia ad avere una forma più allungata; tutto questo la porta ad essere maggiormente confondibile con un essere umano.

5.2.3 Esperimenti con matrici 2×3

In questa classe di esperimenti, due oggetti collidono e dopo un certo tempo escono tre oggetti. La matrice ci dirà chi è il nuovo nato (se farà due scelte) o ci dirà di quali oggetti in input ha perso traccia.

- **Esperimento 2x3.A**

É stato preso un caso di media difficoltà del tipo: $ABC \Rightarrow A'B'$. In particolare l'oggetto C è un'automobile, mentre A e B sono due persone. Il sistema dovrà individuare con facilità l'intruso e potrà distinguere anche tra oggetti A e B.

Le matrici al primo frame valgono:[10] vedi foto fig. 5.5

$$\mathcal{M}_S(0) = \begin{bmatrix} \mathbf{43,81} & 24,20 & 31,99 \\ 15,04 & \mathbf{71,01} & 13,95 \end{bmatrix} \qquad \mathcal{M}_{NS}(0) = \begin{bmatrix} \mathbf{42,89} & 39,03 & 18,08 \\ 6,16 & \mathbf{89,27} & 4,57 \end{bmatrix}$$

Mentre le matrici medie valgono:

$$\overline{\mathcal{M}}_S = \begin{bmatrix} \mathbf{51,4} & 21,7 & 26,9 \\ 13,0 & \mathbf{69,6} & 17,4 \end{bmatrix} \qquad \overline{\mathcal{M}}_{NS} = \begin{bmatrix} \mathbf{71,2} & 22,4 & 6,4 \\ 10,3 & \mathbf{85,5} & 4,2 \end{bmatrix}$$

Al primo frame le matrici sono entrambe coerenti in senso stretto e l'algoritmo di scelta farebbe la stessa scelta (corretta): associare A con A' e B con B'.

Vediamo dunque i parametri:

Parametri per l'esperimento 1 (100 frame)					
stazionarietà	$\overline{C_l\%}$	$\overline{C_s\%}$	$\beta(0)$	$\overline{\beta}(\%)$	R_a %
✓	100,00	97,00	43,35	46,17	97
–	97,00	97,00	62,07	77,45	97

I dati possono essere letti così: il 97% delle volte esiste una scelta ottima con tutti i termini sopra soglia, e che porta alla scelta giusta. In particolare quella scelta vale il 46% (o il 77% nel caso non stazionario!) tra le 6 scelte possibili.

5.3 Distanze a match singolo

In questa categoria ricadono esperimenti 'unari' di matching tra due entità: sequenza-immagine (DSI) o sequenza-sequenza (DSS).

Le DSI si occupano del problema: *"Quanto l'immagine $\mathcal{I}_{target}$ è vicina alla sequenza $\mathcal{S}$?"*. Di solito questo problema si pone quando un oggetto non sia più tracciabile per un certo periodo $\triangle t$ e ci si chiede se $\mathcal{I}$ appartenga o meno alla sequenza.

[10] Si noti che la somma su ogni riga non vale 1 ma 100. É stato fatto per una maggiore leggibilità.

Le DSS si pongono lo stesso problema ma in output non usano l'ultima immagine disponibile, bensì tutta la sequenza che si è formata dall'inizio. Quest'approccio è certamente più completo, ma anche molto 'mediato' (dà un'informazione poco variabile nel tempo, poiché a ogni passo il cambiamento delle statistiche è sempre più basso).

Negli esperimenti svolti sequenze lunghe sono state spezzate (500 frames) in altre più corte (circa 100 frames) ed è stata confrontata una singola sequenza con ogni singola immagine di un'altra, così da avere un campione abbastanza rappresentativo.

La stessa sequenza è stata poi confrontata con oggetti simili e dissimili (per vedere se la distanza riesce a discriminare con qualcosa di simile e se riesce a considerare *molto* diverso qualcosa per cui effettivamente il problema appaia ai nostri occhi banale).

Si noti che il sistema presenta una forte asimmetria tra input e output: essa nasce dal fatto che i primi hanno – al momento dello studio – una storia completa e statica, i secondi una storia inizialmente nulla e in crescita.

Sono state confrontate 10 distanze che si sono mostrate buone nel giudicare la somiglianza tra oggetti; in realtà, la versione stazionaria e non di 5 distanze:

Distanza simmetrica (DSS) Corrisponde alla distanza SS_2 (simmetrica) di pag. 60. In seguito verrà chiamata d_1.

Distanza a eccedenza (DSI) É la distanza a soglia a valor fisso della Sez. 4.4.3, pag. 55. Verrà chiamata d_2.

Distanza pesata (DSS) Non è altro che la distanza SS_3 (pesata) di pag. 61. Verrà chiamata d_3.

Distanza a soglia intera (DSI) É la distanza a soglia a valor fisso della Sez. 4.4.2, pag. 54. Verrà chiamata d_4.

Distanza euclidea (DSI) É la distanza psuedo-euclidea della Sez. 4.4.1, pag. 53. Verrà chiamata d_5.

Come si può notare, vengono trattate allo stesso modo distanze di tipo DSS e DSI. L'approccio misto è reso possibile dalla somiglianza tra

esperimenti (e dall'omogeneità dell'interfaccia): in entrambi i casi viene calcolata la history della sequenza in input una volta per tutte; viene poi fatto un **ciclo** su dati che dipendono dal tipo di oggetto in output (la sequenza in output abbia lunghezza T):

◦ Nel caso di una DSS, vengono fatti T esperimenti con le T sottosequenze ottenute prendendo i frame da 0 a τ (con $\tau = [0, T-1]$); questo simula un comportamento reale.

◦ Nel caso di una DSI, vengono fatti T esperimenti con la τ^{ma} immagine della sequenza di output. Quest'informazione però non tende a convergere a un valore fisso ma piuttosto a distribuirsi secondo una 'nube' che ne indica la *tendenza*; per ovviare a questo fatto questi valori vengono mediati nell'intervallo $[0..\tau-1]$.

Si ricordi che i T esperimenti fatti vogliono simulare la risposta che il sistema darebbe all'istante τ (usando al meglio *tutte e sole* le informazioni raccolte fino a quel momento). Questo artificio è volto a misurare, oltre alla bontà di una distanza, la sua stabilità nel tempo e l'eventuale convergenza a valori esatti dopo un certo numero di frame; così si potrà meglio fare una stima dei tempi di decisione.

Per ogni distanza è stato calcolato un valore cui è stata applicata una soglia. I valori di soglia usati sono:

Soglie per le distanze usate		
Tipo di distanza	Stazionarietà	
	✓	–
Distanza simmetrica (DSS)	15	20
Distanza a eccedenza (DSI)	2,5	4
Distanza pesata (DSS)	0,28	0,28
Distanza a soglia intera (DSI)	18	23
Distanza euclidea (DSI)	10	16

Data una distanza e la sua soglia, si ha in mano una **funzione di decisione** che dice se due oggetti sono *uguali* o *diversi* (non è stata ammessa una zona di indecisione). Per ogni funzione distanza usata d_k, indicheremo con δ_k la funzione booleana di decisione, che assumerà valori diversi a seconda che si tratti di una DSI o di una DSS.

◦ Nel caso di una DSS, i valori sono: **0** se i due oggetti vengono visti *uguali*, **1** se *diversi* (con una semantica di *distanza*, dunque, non di *matching*).

◦ Nel caso di una DSI, i valori vengono calcolati **per ogni istante** allo stesso modo che per una DSS (**0** se i due oggetti vengono visti *uguali*, **1** se *diversi*), ma la distanza al tempo τ (come già detto) viene *mediata* su tutti gl'istanti fin ora calcolati. Anche qui, il valore totale va sempre da zero (se c'è sempre stata uguaglianza) a uno (se c'è sempre stata diversità) ma non è necessariamente intero.

Sono poi state create semplicissime due funzioni di matching 'incrociate', una basata sul responso delle distanze migliori (f_{best}), una invece globale (f_{all}). Noti che siano input e output, esse sono definite come:

$$f_{best} \doteq \delta_1^{NS} + \delta_3^{NS} + \delta_5^{S}; \tag{5.1}$$

$$f_{all} \doteq \sum_{i=1}^{5} \left(\delta_i^{NS} + \delta_i^{S} \right) \tag{5.2}$$

Si noti che la prima ha valori nel range $[0, 3]$ ed è stato usato come spartiacque il valore mediano: essa assume la diversità tra oggetti se e solo se vale $d > 1,5$. La seconda invece assume valori in [0,10]; si è deciso di considerare diversi due oggetti se f ha valori in (5..10] (si è adottata la mediana come valore di uguaglianza: infatti in casi non ambigui i valori sono ben lontani dalla mediana, in casi ambigui studiati conviene avere la mediana dalla parte degli 'uguali').

Con queste premesse, siamo in grado di osservare in ogni esperimento l'andamento delle 10 distanze e delle 2 funzioni di decisione a ogni passo; sono stati calcolati valori in 4 tempi diversi: 1 (calcolo 'a bruciapelo'), 5 (pochi dati), 25 (un secondo di sequenza, eppure già una buona statistica), e il massimo possibile, ovvero la lunghezza effettiva della sequenza (in quest'ultimo caso, spesso, i valori della sequenza in output sono ben lontani dai suoi valori iniziali, e quindi è facile che non diano risultati attesi).

- **Esperimento A**

Questo caso è particolarmente interessante poiché è una *cross-reference* con l'esperimento a pag. 78 (è in realtà l'unico abbinamento lasciato in sospeso dall'algoritmo di matching). I due oggetti in questione sono due versioni (vedere

(fig. (5.1), pag. 78), figure C e C') della stessa automobile viste da angolazioni diverse e a una diversa distanza. I dati sono:[11]

decisioni per l'esperimento A				
frame elaborati	# diversità		uguaglianza *all-in-all*	
	$f_{all}[0..10]$	$f_{best}[0..3]$	f_{all}	f_{best}
1	9	2	–	–
5	6	1,6	–	–
25	4,4	0,12	✓	✓
300	3,39	0,02	✓	✓

In entrambi i casi, la risposta è *definitivamente*[12] corretta, e l'istante di 'convergenza' è $t = 7$.

Questo vuol dire che il sistema considera i due oggetti diversi al primo frame elaborato, e così per pochi frame. Da quel momento, li considererà sempre uguali.[13]

- **Esperimenti B C D E**

In questi quattro esperimenti sono state confrontate 4 sequenze uguali a due a due di persone diverse (una donna e un uomo), analogamente a un caso di tipo: $\boxed{AB \Rightarrow A'B'}$. Verranno mostrate dunque 4 tabelle per le distanze ottenute nei 4 abbinamenti possibili: $\boxed{A \Rightarrow A'}$, $\boxed{A \Rightarrow B'}$, $\boxed{B \Rightarrow A'}$, $\boxed{B \Rightarrow B'}$. Ci si attende un match molto buono tra oggetti uguali e un match più faticoso per oggetti diversi (sono infatti persone abbastanza simili).

[11] sono stati enfatizzati i dati attesi (corretti) nelle ultime due colonne.

[12] nel senso matematico del termine, ovvero esiste un istante tale che da quel momento in poi varrà sempre una certa proprietà.

[13] poiché l'informazione è monotona crescente con il numero di frame elaborati, è buona cosa che la scelta non venga cambiata più avanti nel tempo.

decisioni per l'esperimento B $\boxed{A \Rightarrow A'}$				
frame elaborati	# diversità		uguaglianza *all-in-all*	
	$f_{all}[0..10]$	$f_{best}[0..3]$	f_{all}	f_{best}
1	0	0	✓	✓
5	0,4	0	✓	✓
25	0,72	0	✓	✓
100	0,47	0,02	✓	✓

decisioni per l'esperimento C $\boxed{A \Rightarrow B'}$				
frame elaborati	# diversità		uguaglianza *all-in-all*	
	$f_{all}[0..10]$	$f_{best}[0..3]$	f_{all}	f_{best}
1	10	3	—	—
5	8	2	—	—
25	8	2	—	—
48	6,98	1,98	—	—

decisioni per l'esperimento D $\boxed{B \Rightarrow A'}$				
frame elaborati	# diversità		uguaglianza *all-in-all*	
	$f_{all}[0..10]$	$f_{best}[0..3]$	f_{all}	f_{best}
1	10	3	—	—
5	8	2	—	—
25	8	2	—	—
100	8	2	—	—

decisioni per l'esperimento E $\boxed{B \Rightarrow B'}$				
frame elaborati	# diversità		uguaglianza *all-in-all*	
	$f_{all}[0..10]$	$f_{best}[0..3]$	f_{all}	f_{best}
1	0	0	✓	✓
5	0	0	✓	✓
25	0,4	0	✓	✓
48	1,48	0,02	✓	✓

In questi esperimenti, il sistema è in grado di fornire già al primo frame la risposta corretta.

- **Esperimento F**

In questo esempio i due oggetti sono diversi (un furgone e un'automobile rispettivamente). La after-sequence è molto lunga (398 frame).

decisioni per l'esperimento F				
frame elaborati	# diversità		uguaglianza *all-in-all*	
	$f_{all}[0..10]$	$f_{best}[0..3]$	f_{all}	f_{best}
1	10	3	—	—
5	10	3	—	—
25	10	3	—	—
398	9	3	—	—

- **Esperimenti G H I J**

In questi quattro esperimenti (analoghi agli esperimenti B C D E come struttura) è stato ripetuto il caso 2x2.A (pag. 81), che è particolarmente problematico poiché le immagini da confrontare sono *molto* simili. É stato riportato solo il primo merge-split per brevità: i risultati nel secondo sono analoghi (e meno significativi, in quanto le sequenze sono molto corte).

decisioni per l'esperimento G $\boxed{A \Rightarrow A'}$				
frame elaborati	# diversità		uguaglianza *all-in-all*	
	$f_{all}[0..10]$	$f_{best}[0..3]$	f_{all}	f_{best}
1	0	0	✓	✓
5	0	0	✓	✓
23	1,48	0,43	✓	✓

decisioni per l'esperimento H $\boxed{A \Rightarrow B'}$				
frame elaborati	# diversità		uguaglianza *all-in-all*	
	$f_{all}[0..10]$	$f_{best}[0..3]$	f_{all}	f_{best}
1	5	1	✓	✓
5	4,8	1	✓	✓
23	5,09	1	—	✓

decisioni per l'esperimento I $\boxed{B \Rightarrow A'}$				
frame elaborati	# diversità		uguaglianza *all-in-all*	
	$f_{all}[0..10]$	$f_{best}[0..3]$	f_{all}	f_{best}
1	3	1	✓	✓
5	1	0	✓	✓
23	0,57	0	✓	✓

decisioni per l'esperimento J $\boxed{B \Rightarrow B'}$				
frame elaborati	# diversità		uguaglianza *all-in-all*	
	$f_{all}[0..10]$	$f_{best}[0..3]$	f_{all}	f_{best}
1	0	0	✓	✓
5	0	0	✓	✓
23	0	0	✓	✓

Negli esperimenti H e I, il sistema sbaglia sempre la risposta – ad eccezione della f_{all} nell'esp. H che sembra indecisa e verso fine sequenza (dal frame 16 all'ultimo) trova A e B diversi. Questo era attendibile, poiché sono state confrontate sequenze *molto* simili sotto ogni profilo. Magari un set di descrittori più mirato avrebbe potuto catturare meglio le differenze.

5.4 Conclusioni sugli esperimenti svolti

Le matrici di matching sono atte a risolvere problemi complessi in cui entrino più oggetti, quindi ad affrontare *in toto* un problema di merge-split. Le distanze a match singolo, viceversa, offrono un risultato di tipo diverso: quanto due oggetti possono essere ritenuti uguali? É facile pensare a una possibile integrazione tra due strumenti. Parte degli esperimenti fatti sembrano dimostrare che:

Un approccio *ibrido* tra distanze a match singolo e matrici di matching è la scelta migliore: le prime sono più forti a identificare in assoluto analogie e differenze tra oggetti, ma inevitabilmente falliscono quando questi siano troppo simili; le seconde hanno un minore potere descrittivo ma avendo un'idea di *contesto* riescono a fare scelte più 'intelligenti' nel caso di fenomeni con più oggetti.

Nulla peraltro vieterebbe di fare esperimenti con matrici di matching le cui celle vengano riempite con una delle 10 funzioni distanza studiate (o un loro *blending*).

5.5 Stima sull'efficienza dell'estrattore di *features*

In questa sezione si tratterà del problema dei costi computazionali del sistema al fine di avere stima dell'incidenza del modulo su un generico sistema di tracking. Si parlerà di costi in termini di *tempo*, non di *spazio*, poiché l'occupazione di memoria non costituisce un vincolo significativo sui calcoli (per problemi comuni, almeno).

Poiché tutto il lavoro è stato svolto off-line, possono essere fatte solo delle stime sull'impatto del modulo di estrazione features (che dovrebbe contribuire alla solidità del modulo di Tracking). Certamente, un occhio di riguardo è stato posto sul costo computazionale; tracking on-line e tracking off-line, proprio per i diversi vincoli che impongono, possono essere considerati due branche diverse della stessa disciplina:

- **Tracking on-line.** Poiché il vincolo è di elaborare N frame (tipicamente dai 10 ai 25) al secondo, si ha a disposizione meno di un decimo di secondo (allo stato attuale dell'arte – inizio 2002 – circa 10^7 operazioni) per elaborare *sotto ogni profilo* l'immagine, quindi aggiornare il background, estrarre i blob, identificare gli oggetti, e infine tracciarli. Ciò impone moduli snelli di estrazione dell'informazione in ogni parte della catena.

- **Tracking off-line.** Talvolta è interessante documentare (in modo automatico) fenomeni d'interazione tra oggetti su una vecchia sequenza senza limiti stringenti di tempo. Le tecniche adottate sull'estrazione di sfondo (e, nel nostro caso, di features) e le euristiche (non si trascuri che si può tornare indietro sulle scelte e magari raffinarle!) sono molto più pregiate e, per quel che riguarda l'estrazione dello sfondo, si è praticamente arrivati ad una conoscenza *perfetta*.

Il sistema sviluppato è stato inteso per essere inserito in un sistema di *Tracking on-line*, e quindi per essere (o *poter* essere) leggero. Questo offre un'ulteriore giustificazione alla scelta di usare una generica n-pla di descrittori senza la ricerca di una *rosa perfetta*: tutto questo al fine di poter aggiungere e togliere anche secondo criteri di costi. Si potrà prendere un insieme minimo per casi on-line con immagini grosse [14], e un insieme più grosso nel caso non vi siano limiti di tempo. Tanto per fare un esempio estremo, in [9] vengono addirittura calcolate 30 matrici di co-occorrenza (per distanze che vanno da 1 a 30) e si estraggono le features vere e proprie da quella che presenta il miglior comportamento secondo criteri statistici di classificazione[15].

La suddivisione dei descrittori nel Cap. 2 è stata fatta seguendo un semplice criterio che torna utile in questo momento, ovvero lo *strumento* che occorre usare per calcolare un certo descrittore; ci sono infatti gruppi di descrittori che richiedono il (costoso) calcolo di una matrice, dopodiché possono essere (facilmente) ricavati tutti. Ebbene, se la suddivisione in famiglie è fatta con questo criterio, si può osservare con maggiore facilità la motivazione del calcolo di descrittori 'deboli' a favore di una maggiore efficienza di calcolo.

Vediamo di stimare i costi delle seguenti famiglie con riferimento all'implementazione attuale. Si useranno le seguenti definizioni: X (larghezza della bounding box), Y (altezza), $\mathcal{A}$ (area), N_g (numero di grigi considerati – la massima risoluzione nelle immagini usate è 256). *Non* necessariamente N_g dovrà coincidere per istogrammi, matrici di Haralick o vettori di Unser.

Momenti. (Sez. 2.4.1) Il calcolo dei momenti, per efficienza, viene fatto in contemporanea per tutti i momenti di ordine fino a 3 (per un totale di 16 momenti primitivi, da cui si possono estrarre i centrali e i normali). É stato usato un algoritmo della CV [1], ma senz'altro il costo deve aggirarsi intorno a $C = \mathcal{O}(\mathcal{A})$.

Contorni. (Sez. 2.4.3) Questa famiglia richiama un algoritmo della CV [1] al

[14] Il costo di elaborazione risente molto della dimensioni di un'immagine: non è facile stimare la *complessità* poiché nella pipeline vi sono algoritmi di costo diverso. Con buona approssimazione si può pensare a $C(A) = \mathcal{O}(A^\alpha)$, con $\alpha \in [1, 2]$, dove A è l'area del frame.

[15] si cerca, con una misura di tipo χ^2, la matrice con righe e colonne maggiormente scorrelate tra loro.

cui costo non è facile risalire (non è documentato e varia secondo molti parametri). In generale esso trova tutte parti connesse e le esplora marcando il perimetro con un numero pari all'etichetta della parte connessa in questione. Il costo di ricerca può essere approssimato con una somma di una ricerca su area più un fattore dipendente dal numero di contorni trovati e dalla lunghezza. Una stima molto approssimativa può essere: $\mathcal{O}(\mathcal{A}) < C < \mathcal{O}(\mathcal{A}^2)$.

Istogrammi. (Sez. 2.5.1) Il costo dell'algoritmo di generazione dell'istogramma è *indipendente* dalla *grana* (intesa come grossezza di un *bin*). Non vale lo stesso per l'occupazione di memoria. Il costo è sempre $C = \mathcal{O}(\mathcal{A})$.

Matrice di Co-occorrenza. (Sez. 2.5.2) Il costo di riempimento della matrice è $C_{fill} = \mathcal{O}(\mathcal{A})$ e non dipende quindi da N_g ; con buona approssimazione, ognuna delle 14 features richiede di vagliare o l'intera matrice ($C_{feat_K} = \mathcal{O}(N_g^2)$) o un vettore estratto da essa ($C_{feat_K} = \mathcal{O}(N_g)$). Trascurando i secondi, il costo totale sarà del tipo $C = \mathcal{O}(\alpha\mathcal{A} + \beta N_g^2)$. Occorre stimare i due coefficienti per vedere quale fattore incida di più. Si possono usare ($\alpha = 8$;[16] $\beta = 11$([17])). Si ricorda che vengono estratte 4 matrici e dalla matrice-media si calcolano vettori e parametri.

Vettori somma e differenza. (Sez. 2.5.2) Ancora una volta, il costo di riempimento di entrambi i vettori è pari a quello di scansione dell'immagine: $C_{fill} = \mathcal{O}(\mathcal{A})$. Per quel che riguarda le features, il costo per ciascuna è $C_{feat} = \mathcal{O}(N_g)$. Il numero di features estratte è $2 \cdot 4 \cdot 3 = 24$ (due istogrammi, 4 features, 3 tipologie d'istogramma – orizzontale, verticale e mediato). Il numero d'istogrammi calcolato è di 5 coppie. Qui potremmo avere: $C = \mathcal{O}(8\mathcal{A} + 24N_g)$. Il vantaggio di Unser è il rendere lineare (da quadratico) il costo di estrazione delle features, oltre a diminuire drasticamente l'occupazione di memoria.

[16] 8 vicini confrontati a ogni passo.

[17] 7 parametri che richiedono un ciclo doppio sulla matrice più calcolo di 4 vettori – che richiedono anch'essi N_g^2 calcoli.

Altro. Le features non citate sono supposte avere un costo trascurabile (nessun ciclo).

Il sistema è stato fatto funzionare su un computer Pentium III (Intel), MS Windows 98, 1000 GHz, 256 MB di RAM; il calcolo di descrittori (con comprensiva matrice) per sequenze di lunghezza variabile hanno portato ai seguenti tempi:

Stima dei tempi per sequenze				
nome sequenza	$\mathcal{A}_{BB}$	# frame	$\triangle t_{tot}$ [sec]	τ_{frame} [ms]
persona 1	500	100	1,600	16
persona 2	2500	94	2,200	23
persona 3	155	1460	26,030	17
veicolo 1	4500	82	3,520	42
veicolo 2	11400	102	9,560	94

Il tempo medio di elaborazione di una sequenza è di $30\,ms$. Sono state presentate le prestazioni per un eseguibile compilato in *modalità DEBUG* [18] e accedendo alle immagini da disco (non da memoria). Si ritiene di poter migliorare le prestazioni in un sistema di Tracking on-line di almeno un fattore 2 rispetto ai valori dati.

Già ora, affiancando il modulo a un programma di tracking che abbia circa la sua stessa complessità temporale di elaborazione (30 ms a frame), e supponendo di avere 3 oggetti in esame a ogni frame, la singola elaborazione costa circa $30 + 3 \cdot 30ms.$, sopportando un *throughput* di *almeno*[19] 8 frame al secondo, non infrangendo di troppo il vincolo di real-time. Semplici ottimizzazioni (per esempio calcolando i descrittori più costosi solo a certi intervalli) possono poi elevare ulteriormente la velocità media.

[18] non *release*, per problemi con alcune librerie.

[19] Tutte le stime son state fatte in difetto .

Capitolo 6

Conclusioni

> The most beautiful thing we can experience is the mysterious. It is the source of all true art and science. He to whom this emotion is a stranger, who can no longer pause to wonder and stand rapt in awe, is as good as dead: his eyes are closed.
>
> *Albert Einstein*

6.1 Introduzione

In questo capitolo verranno tirate le somme del lavoro svolto; si è ritenuto importante, tuttavia, suggerire alcune possibili direzioni in cui muoversi ai fini del proseguimento di questa attività di ricerca.

Nella sezione che segue (Sez. 6.3) vi è una *summa* del lavoro svolto, mentre per i possibili sviluppi si rimanda alla Sez. 6.2.

6.2 Possibili estensioni e sviluppi futuri

Il sistema è fondamentalmente un prototipo funzionante che si presta a sviluppi in molte direzioni.

Colori

Le immagini bitmap usate sono state acquisite a 256 livelli di grigio. Una necessaria estensione è la possibilità di gestire i colori; poiché questi sono rappresentati tramite tre canali (RGB, CMY, ...) *ciascuno dei quali* è del tutto simile a un gray-level è possibile applicare gli stessi discorsi già fatti triplicando il numero di descrittori[1] (o cercando un qualche nuovo parametro che sia magari indicativo del colore nel suo insieme); in particolare, il colore rende possibile individuare le ombre e quindi consente una migliore segmentazione delle immagini (meno rumore). Proprio per questo, gli esperimenti di tracking vengono *tipicamente* divisi in categorie diverse a seconda che i dati siano a colori o a toni di grigio.

Influenza del real-time sulla scelta della stazionarietà dei modelli.

Se il modulo di alto livello ci dice l'*età assoluta* delle sequenze, nel confrontare due sequenze possiamo decidere se usare il modello stazionario o meno a seconda dei tempi di vita *assoluti* delle due sequenze da confrontare. Si è sperimentato infatti un comportamento diverso dei due modelli a seconda che l'after-image sia più o meno recente; spesso il caso non stazionario è migliore ma quando la fase di congelamento (*freeze*, pag. 58) è lunga diventa più potente il metodo stazionario (poiché non ha più senso pesare maggiormente immagini più recenti nella prima sequenza quando queste siano tutte molto lontane dalla seconda).

Viene qui proposto un semplice algoritmo di decisione del parametro (stazionario o meno) ammesso che venga fornita un'informazione sull'età assoluta (e quindi del loro scostamento relativo) degli oggetti.

Se per esempio occorre confrontare una sequenza lunga τ con un'immagine nata $(t+1)$ immagini dopo si può fare un *blending* tra i valori stazionario o meno basato su quanto l'immagine sia 'obsoleta' (δ alto) o 'recente' (δ basso). Siano κ_S e κ_{NS} rispettivamente il parametro stazionario e non. I casi limite devono valere:

$$\begin{aligned} \lim_{t\to\infty} f(t) &= \kappa_S; \\ f(0) &= \kappa_{NS}; \end{aligned}$$

[1] ammesso che ciò abbia senso: i descrittori di maschera non fanno leva sull'informazione 'colore' quindi non cambierebbero per niente nel caso a colori.

Una possibile funzione di blending esponenziale può essere:

$$f(t) = \kappa_S + \alpha^{t/\tau}(\kappa_{NS} - \kappa_S), \tag{6.1}$$

dove α è un opportuno parametro adimensionale (si consiglia un valore basso, tipo $0,2$), che contribuisce a esprimere il 'tempo di dimezzamento' della funzione.[2]

Classificazione

Il modello gaussiano a N descrittori si presta molto bene a una classificazione. In generale, ogni immagine avrà associato ad essa un punto in uno spazio a N dimensioni; ammesso che si decidano dei gruppi statici di appartenenza, possibilmente dicotomici (persona - non persona, singolo - gruppo, veicolo - non veicolo[3]), è possibile tracciare in modo semplice degli iperpiani che taglino lo spazio in due nella speranza di isolare da un lato tutti gli oggetti appartenenti alla classe e dall'altro quelli non appartenenti.

Un ulteriore modo per giudicare il sistema stesso dei descrittori può essere quello di vedere quanto questa classificazione sia efficace in casi reali.

6.3 Conclusioni

É stato proposto un modello (e una relativa implementazione) di caratterizzazione delle immagini tramite *appearance models* che si è mostrato in grado di:

- risolvere correttamente semplici problemi di tracking associando correttamente sequenze pre-*merge* a immagini post-*split*;
- affrontare correttamente problemi più complessi (numero maggiore di oggetti, somiglianza tra oggetti diversi, diversità tra oggetti 'uguali');
- risolvere correttamente questi ultimi problemi nell'ipotesi di 'mondo chiuso'[4], ovvero nell'ipotesi che nessuno nasca o muoia e che quindi vadano

[2] Per la precisione questo è proporzionale a $\tau \cdot \log(\alpha)$.

[3] addirittura si può pensare alla divisione persona-veicolo, fatte opportune ipotesi sul dataset.

[4] La *Close World Assumption* è stata presa in prestito dall'Intelligenza Artificiale. Nel suo contesto originario allude alla forte ipotesi che: "tutto ciò che si vede è tutto cio che effettivamente *è* nel sistema".

scelte solo le opportune associazioni. Quest'ipotesi è forte ma spesso più che ragionevole;

- riconoscere uguaglianza o diversità tra due oggetti, dimostrando una buona robustezza anche nel caso di stravolgimenti dell'immagine stessa (cambio di prospettiva, *scaling*, ...).

É stato affrontato il problema in maniera **modulare** (nel senso dell'Ingegneria del Software, quindi sostanzialmente da un punto di vista di *progetto* e *implementazione*) che rende il sistema in grado di:

- adattarsi in futuro a esigenze di **real-time**, scegliendo la n-pla di descrittori più confacente a tali esigenze (compromesso tempo - capacità di descrizione);
- supportare i **colori** in nuove *release*; con pochissima fatica si può estendere il sistema facendo in modo di avere un'informazione tripla per quel che concerne istogrammi e matrici di co-occorrenza. Questo rende sicuramente molto più forte la caratterizzazione di oggetti;
- interfacciarsi a moduli di basso livello in real-time [5];
- fornire un servizio fruibile da un eventuale modulo di più alto livello che prenda decisioni più 'pregiate' avendo una maggiore conoscenza contestuale ma sfruttando le informazioni fornite da questo livello.

[5] l'interfaccia è molto semplice e accetta in ingresso semplici puntatori a immagini della libreria Ipl [2] che è uno standard molto diffuso; non importa dunque se vengano da file o da memoria né importa il loro contesto. Il sistema funge da semplice servitore.

Ringraziamenti e saluti

L'ingegnere non vive, *funziona*.

Anonimo

Introduzione

In questo capitolo verranno resi ringraziamenti alle persone che ho in mente in questo momento; poiché la tesi è un pelino stressante, non me ne vogliano coloro che non trovano il loro nome qui. Saranno ripagati opportunamente con nettare di Bacco alla prima occasione.

Si noti che dai ringraziamenti si glissa sui saluti; come dice il mitico Giulio Cesare Barozzi, l'appetito vien mangiando (e anche *Io i miracoli li faccio solo di Mercoledì*, e oggi *è* mercoledì!).

Ringraziamenti veri e propri

L'idea di lavorare di cut-n-paste da una tesi altrui mi era effettivamente balenata in testa, ma avevo poi paura di trascurare di cambiare qualche nome e ho optato per una cosa fatta tutta da me.

Comincio col ringraziare i miei (mamma Lucilla e papà Bruno) per la pilla e la pazienza messe in questi anni di studio; i parenti più vicini: sorella Ele per avermi insegnato la pazienza (con lei ce ne vuole tanta, e coi pargoli ancor di più); il marito (suo non mio) JC per averla imparata da me (arriva prima o poi quella birra, tranquillo!); i nonni che da quando sono stati introdotti gli euro hanno raddoppiato le loro mancette: viva gli euro!!! Il prossimo 'pangeo' lo potete fare da 5000 £?!? Così divento miliardario...

Un altro ringraziamento va sicuramente alla mia Crysania (un salutone anche a genitori, nonne, zii e cugina!) perché a momenti le spunta un'aureola con 4 ali (e per citare qualcuno, *chi mi conosce lo sa*), altro che Lines Seta: grazie per avermi insegnato gli accostamenti dei colori, per aver sopportato le mie elucubrazioni matematiche e per avermi fatto scoprire un intero mondo di curve trigonometriche fuori da un libro di analisi. Un bacio grosso così: 10^{42}.

Che dire poi di Ridge che si è laureato il 20-3-2002 (dovrei dire oggi?) con me solo per farmi compagnia?

Un saluto a Pierandrea cui dedico una formula fisica che a lui piace tanto (poi te la spiego): $E = mc^2$. So che il LaTeX può far di meglio ma non c'ho tempo, scusa. Un bacio anche alla Marci.

Un saluto a Gigi il tirapacchi che già prima non si vedeva, e ora che sta a Bologna figurarsi.

Un saluto a Giuà (who doesn't die, you re-see him) e ai genitori e già che ci siamo a tutta la compagnia che mi ha fatto conoscere (Laco e amante, Vench, Rocchi, Alice, Eleonora, Manu, Zerbo, Ciccio[6] e Lollo).

Un bacione a tutte le donne che conosco (amanti comprese).

Un bacione a Mino e Sara. Complimenti Mino, ora anche tu *navighi* in Internet.

Una baco (di windows) particolare al Club Seghe Mentali, in particolare a Nikotra, Zerbinus, Vincenzo, e Carlo Perassi che si è addirittura letto la mia tesi! Spero che i vostri propositi di borchiare dei Chiorboloni per la *Crucifixio Carlessi* siano andati in porto!

Un saluto particolare a Titti che è sempre tanto caro (a cui dedico una devota minzione)! Scusa se ho scritto la tesi sotto Winzoz, non lo faccio più, giuro. Ma ho usato perlopiù materiale opensource!!!

Un bacio ai miei amici di Argenta: Gigo, Condor, Rinaldi e tanti altri. Un saluto anche ai ragazzi dei GdR (o RpG?): Bocia, Tacco, Muzza (raditi!).

Un salutone al pub la Mona Nera e Franco e la Marilù che mi hanno fatto per la prima volta sentire a casa mia in un pub! Anche se quegli short fanno davvero male...

[6] ogni compagnia ha il suo Ciccio, ci avete mai fatto caso? E spesso son pure magri!

Un saluto ai chattari: Mad, Atu, Scarlet, Anto, il Corvetto (sei troppo cartola!), Emp.

Un grosso grazie ai Gem Boy perché sono semplicemente dei grandi, e così matti da avermi fatto suonare con loro Trenino (che performance, hanno riso più in quei 5 minuti che in tutto il resto del concerto... scritturatemi!!! Racconto anche barzellette peggiori di quelle di Siro!).

Un saluto e un caloroso ringraziamento a Max Ferri (il nostro Santone[7]) che mi ha ispirato la faticosa e non redditizia via dell'insegnamento, a Martino (per quel poco che ci siamo visti!) e a Di Stefano che è stato una buona guida per la tesi[8]. Altro dir non vo', capirete il perché.

Un saluto all'appartamento: Stefanone, Roby e Luca. Un ciao anche a Cricca e Cris (e morosa). E i compagni di corso? Soccia ce n'è troppi... Pivot, Dave, Berto (e con lui tutto il mio ex gruppo!), Senegal e Francesco, il mitico Baietti, Lorenzo, Maurizio, Buffaldini, Bonolis, Varietà, la bella Emilia e quel cartolone arrapato di Mauro Costa Bizzarri. Un grosso saluto a quel ravennate di Vector Field alias Enrico 'Punk' Biondini.

Un bacio anche alla mia Sarah e a tutta la sua balotta (Frascé, Micky, il piccolo grande Piolo).

Diamo inizio alla sfilza di goliardi:

Un bacione grosso a tutta (o quasi) la mia Balla: Hak (sei un grande!), Ramanta (attenta alle ragnatele, t'aiuto io se vuoi), Scarpe (mi regalerai un paio di mutandine?), la Ià, Thierry, Nik, Canguria, Spaniza, Mementula (bella gn***a!) e il mitico Carlo (ma non la sua amica, quella del timpano x intenderci). Un bacione speciale a Incompiuta e Holyday (ma quando ti farai vedere?). E ricordate: *Stare in montagna può anche talvolta ottenebrare ragionevoli nordisti. Anche tu evidentemente non educhi a cullare atavici sentimenti. Avveditene.*

Un saluto ai vecchioni: il Nasone, Pierpaolo e Scazzius, Serpicus (mio idolo di forma e stile... che pessimo adepto hai!), l'elegante Violetta e la solare Francesca. Un salutone anche a Fraffo e al logorroico regista Jalbar.

[7]girano voci che uno che manco l'aveva avuto a lezione prima di dare un esame l'avesse cercato per *toccarlo* come portafortuna.

[8]a quel che mi avevan detto altri tesisti, avevo una gran paura di dover riscrivere la tesi daccapo 5 volte!!! E invece no!

Minzione speciale (oserei dire un cospicuo incrocio di flusso) anche per Davide (Celhainmano) che ora fa il pulotto, ma dite voi... che spero mi abbia organizzato un bello scherzone per oggi.

Un salutone a Guido (chatta meno e lavora di più) e alla Ros (lavora di meno e chatta di più!).

Un bacio particolare a Mannarus, Marzia e moroso (per chi non lo sapesse, una coppia di due goliardi di nome Pearl Drops e Colgate, ditemi voi...[9]) e a tutta la Parochia veneta (Palestina, non è che sei gay?) . Già che ci siamo salutiamo anche tutti i nerds (come me) delle 3PT (Treppalle Terûn: il gran calippo, Aquila, Termy e), Ranaplan, Canottiera, Aladano (sei stato il mio modello ed esempio!) e Cippo, Bozen e Athena, Sberla, Brescla, Strafichino Rondolo e Johnny Glamour.

Un saluto anche ad amici esteri:

PG Babi, Isa, Sperella, Eowin, il Decano, Arra, Shirly, il Grifo d'ora Fra Tac (un HC? Fo la collezione!) :-))) e il Grifo d'allora Gattapone, Bimbabella, l'ing. Peregrino Tuc, Galatina, l'ex Califfa e un bacio a tutta la cittadinanza: siete squisiti! Mi sento più a casa mia lì da voi che a Bologna...

FE Gigi Tornello, Adamo e tutti gli scacchi che mi dedicano sempre *Universalità*,siete un tesoro! . Un salutone a Dodipeto e Maoman che includo in FE per motivi storici.

PV Frenulo! Dove sei finito? Ciaaaaaaaaaao!

TO Manolo (di cui porto retaggi organici sul mantello, ormai a TO è leggenda), Lord O, Giuditta, Caramon, Furia, Rulettus (la grolla mi ha cambiato la vita! I've seen the light!) e Coccius(dove sei finito?)

TS Tigre, ma quanto sei gn***a! Torna col tuo ex, però!

PD Seneca e morosa: il regalo di compleanno era galattico, volevo farci le bomboniere ma mia madre (la fautrice) ha posto il veto :-(((

PI Spurgo sei letteralmente un mito! Torna matricola, però!

[9]Come se un Palladius si accoppiasse con una che si chiama Athena...

GE Nep e Torroncino e tanti altri di cui non so il nome: che belli che siete voi genovesi! Proprio bravi!

PR Defensor: ma sei *ovunque*?!?

Un abbraccio forte a chi non ho menzionato: ti *assicuro* che ti avevo sulla punta della lingua.

PS. Anna ti amo.

Riccardo Carlesso,
Argenta (palustre urbe de la bassa de la estense civitate)
lo septimo die de lo terzo menstruo
de lo primo ano *Post Odisseam Spazialem Factam*

Bibliografia

[1] *Open Source Computer Vision Library*, 1999-00 Intel Corporation, http://developer.intel.com/

[2] *Intel Image Processing Library*, 1997-98 Intel Corporation, http://developer.intel.com/

[3] *2nd IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, http://pets2001.visualsurveillance.org

[4] *Handbook of Computer Vision and Applications*, vol. 2: Signal Processing and Pattern Recognition, Academic Press, 1999.

[5] Roß, T., Handels, H., Busche, H., *et al.* (1995). *Automatische Klassifikation hochaufgelöster Oberflächenprofile von Hauttumoren mit Neurolnalen Netzen*. In *Mustererkennung 1995*, G. Sagerer *et al.*, ed. Berlin: Springer-Verlag.

[6] Gonzalez, R. C., and R. E. Woods, *Digital Image Processing*, Addison-Wesley, 1993.

[7] Athanasios Papoulis, *Probability, Random Variables, and Stochastic Processes*, 1965, 2nd edition, 1984, McGraw Hill.

[8] R.M. Haralick, R. Shanmugan, I. Dinstein, *Textural Features for Image Classification*, IEEE Trans Syst. Man Cyb., vol. SMC-3, no. 6, pp. 610-621, 1973.

[9] K.J. Khiani, S.M.Yamany, A.A. Farag, *Classification of the Effects of F-Actin Under Treatment of Drugs in Endothelial Cells*, Computer Vision and Image Processing Lab, Dept of Electrical Engeneering, Louisville, KY 40292.

[10] Unser, M., *Sum and Difference Histograms for Texture Classification*, IEEE Transactions on Pattern Analysis and machine Intelligence, vol. PAMI-8, n. 1, Gennaio 1986.

[11] McKenna, S. J., Jabri, Duric, Rosenfeld, Wechsler, *Tracking Groups of People*, Computer Vision and Image Understanding **80**, 42-56 (2000). Disponibile online: http://www.idealibrary.com

[12] A. Senior, A. Hampapur, Y.L. Tian *et al.*, *Appearance Models for occlusion Handling*, Proceedings 2nd IEEE Int. Workshop on PETS, Kauai, Hawaii, USA, December 9 2001.

[13] M. Hu, *Visual pattern recognition by moment invariants*. IRE Trans. Information Theory, Vol. IT-8, Num. 2, 1962.

[14] T.E. Boult, R.J. Micheals, X. Gao, M. Eckmann, *Into the Woods: Visual Surveillance of Noncooperative and Camouflaged Targets in Complex Outdoor Settings*, Proceedings of the IEEE, vol. 89, no. 10, October 2001.

[15] Arthur E.C. Pece, *Tracking of Non-Gaussian Clusters in the PETS2001 Image Sequences*, Proceedings 2nd IEEE Int. Workshop on PETS, Kauai, Hawaii, USA, December 9 2001.

[16] J.S. Weszka, C. R. Dyer and A. Rosenfeld, *A comparative study of texture measures for terrain classification*, IEEE Transactions on Systems, Man and Cybernetics, vol. 6, no. 4, pp. 269-285, 1976.

[17] M. Mola, *Un sistema di Tracking basato su regole per applicazioni di videosorveglianza*, tesi di Laurea in Ing. Informatica presso l'Università di Bologna; rel: Di Stefano, L.; corr: Mello, P., Viarani, E.; A.A. 2000-01; http://www-labvisione.deis.unibo.it/degree/mmola/degree.html

[18] A. Fabbri, *Analisi di immagini per applicazioni di videosorveglianza: sviluppo di algoritmi di base e valutazione delle prestazioni*, tesi di Laurea in Ing. Elettronica presso l'Università di Bologna; rel: Di Stefano, L.; corr: Neri, G., Viarani, E.; A.A. 1999-2000; http://www-labvisione.deis.unibo.it/degree/afabbri/degree.html

[19] S.U. Thiel, *Object Identification using Texture*, disponibile online: http://www.cs.cf.ac.uk/User/S.U.Thiel/thesis/chapter2_11.html